FDRMOVE

and

FDRERASE

User Documentation

V5.4 L80

THIS PAGE WAS INTENTIONALLY LEFT BLANK

FDRPAS™, FDRMOVE, AND FDRERASE VERSION 5.4

OVERVIEW

PURPOSE OF THE MANUAL

The purpose of this manual is to provide you with the information to install, use, and understand FDRPAS™ (FDR Plug and Swap™), FDRMOVE, and FDRERASE.

WHAT IS FDRPAS™?

This manual documents FDRPAS in Chapter 300. FDRPAS is used to perform a non-disruptive move of a DASD volume from one physical device to another. FDRPAS allows an installation to install new DASD hardware and move existing DASD volumes to the new hardware from older hardware, without disrupting normal operations or requiring a re-IPL. FDRPAS can also be used to move volumes within an existing configuration for load balancing purposes, and to create point-in-time copies of volumes for non-disruptive backups.

WHAT IS FDRMOVE?

This manual documents FDRMOVE in Chapter 325. FDRMOVE combines the proven technology of FDRPAS (volume migration) and FDRINSTANT (instant data set copy) to provide non-disruptive or minimally-disruptive movement of data sets.

FDRMOVE can be used to move data at the data set level to new DASD, and is particularly useful for combining data from smaller DASD volumes onto larger DASD volumes, such as moving three (3) 3390-9s onto a 3390-27 or 3390-54s to a 3390-A Extended Address Volume (EAV). FDRMOVE is very easy to use.

WHAT IS FDRERASE?

This manual documents FDRERASE in Chapter 330. FDRERASE is used to erase all data from DASD volumes before they are reused or removed from a data center. FDRERASE can be licensed as part of your FDRPAS license and can also be licensed separately.

FDRPAS™ is a trademark of INNOVATION DATA PROCESSING and is registered with the US patent office.

FDRPAS™, FDRMOVE, and FDRERASE are proprietary program products of:

INNOVATION DATA PROCESSING
Innovation Plaza
275 Paterson Avenue
Little Falls, New Jersey 07424-1658

and is available for license exclusively from INNOVATION DATA PROCESSING,

Reproduction of this User Manual is prohibited except by licensed users for their internal use.

A copy of this manual is available in Adobe PDF format on the INNOVATION DATA PROCESSING Documentation CD-ROM and on the INNOVATION DATA PROCESSING FTP site for online use and local printing.

© Copyright 2001, 2013 INNOVATION DATA PROCESSING

USA CONTACTS

Corporate Headquarters

Voice	973-890-7300	Sales	sales@fdrinnovation.com
Fax:	973-890-7147	Support:	support@fdrinnovation.com
Home page:	http://fdr.com		

EUROPEAN OFFICES

France		Netherlands (& Nordic Countries)	
INNOVATION DATA PROCESSING		INNOVATION DATA PROCESSING	
191 Avenue Aristide Briand		Brouwerstraat 8	
94230 Cachan		1315 BP Almere	
Tel:	(33) 1 49 69 94 02	Tel:	036-534 1660
Fax:	(33) 1 49 69 90 98	Fax:	036-533 7308
frsales@fdrinnovation.com		nlsales@fdrinnovation.com	
frsupport@fdrinnovation.com		nlsupport@fdrinnovation.com	

Germany		United Kingdom	
INNOVATION DATA PROCESSING International Ltd.		INNOVATION DATA PROCESSING Ltd.	
Orleansstraße 4a		Clarendon House	
D-81669 München		125 Shenley Road	
		Borehamwood, Herts, WD6 1AG	
Tel:	089-489 0210	Tel:	0208-905 1266
Fax:	089-489 1355	Fax:	0208-905 1428
desales@fdrinnovation.com		uksales@fdrinnovation.com	
desupport@fdrinnovation.com		uksupport@fdrinnovation.com	

PRODUCT DEMOS

Product Demos are available on the INNOVATION DATA PROCESSING web site for FDRPAS, FDRMOVE, and FDRERASE. Go to web site http://fdr.com and click the desired link under the “DEMOS” icon.

The direct link for the FDRPAS, FDRMOVE, and FDRERASE Product Demos is: http://www.fdr.com/demo.cfm.

Control Statement Format

Control statements for FDRPAS consist of 80-character logical records ("card images") where:

Columns 1 to 71	Contain the command, operands, and comments fields, except when continued to subsequent logical records.
Column 72	Must be blank if the last character of the command or operand is in column 71.
Column 73 to 80	Not used by FDRPAS, FDRMOVE, and FDRERASE; may contain an identification or sequence number.

General Format

FDRPAS control statements are compatible with those accepted by most other FDR programs, so the format is familiar to FDR customers. The format of control statements is:

```
Command  Operand  Comments
```

where:

Command Field

The COMMAND field identifies the control statement. Each FDRPAS program has a set of commands it accepts, as documented in the rest of this manual. The command can start in column 1 of the input record, or it can optionally be preceded by any number of blank columns, as long as it ends before column 72. It cannot be continued to another input record. It must be followed by at least one blank column.

Operand Field

The OPERAND field, if present, follows the command field and is separated from it by at least one blank. The operand field consists of one or more operands, separated by commas (a common mistake is separating the operands by blanks instead of commas, causing the extra operands to be taken as comments). The operand field may not contain embedded blanks except within quoted strings. Operand fields may be continued onto subsequent logical input records but the first operand must appear on the same record as the command (Continuing an Operand Field).

Most FDRPAS commands require operands. The operands accepted with each command are documented in the rest of this manual.

Operands are keywords, meaning that multiple operands may appear in any order.

Comments Field

The comments field, if present, follows the operand field and is separated by one or more blank columns. It may contain any characters; it is not validated. Comments fields may not be continued, that is, they must end at or before column 71. Comments are not permitted on a control statement that allows operands but on which no operands have been specified.

You can also specify an entire record of comments by placing an asterisk (*) in column 1 of an input record. You can use the rest of the record for comments.

Suboperand

Some operands consist of a list of suboperands. A sub-operand list must be enclosed within parentheses, unless the list reduces to a single sub-operand, in that case the parentheses may be omitted.

Continuing an Operand Field

When the total length of the operands on a statement exceeds the available columns in a logical record, they must be continued onto one or more following logical records.

To continue a statement, interrupt the operand field after any complete operand or sub-operand, including the comma that follows it. That comma must be occur at or before column 71 and the next column must be blank. On the next input record, start the next operand anywhere in the record (columns 1 to 71).

You may continue the control statement any number of times. Some users put only one operand per record to improve readability.

SAMPLE CONTROL STATEMENTS

Here are some examples of properly formatted control statements, including continuation and comments:

```
Column   1         2         3         4         5         6
....+....0....+....0....+....0....+....0....+....0....+....0....
SWAP TYPE=FULL

          SWAPDUMP TYPE=FULL,        FDRPAS SWAPDUMP
                   LARGERSIZE=OK,    OK to use larger target DASD
                   PACING=DYNAMIC    Dynamic Pacing

  MONITOR  TYPE=SWAP,ALLOWPAV=YES         FDRPAS MONITOR
  MOUNT    SWAPUNIT=(3A0*,3B0*,
                3C0*,3D0*,3E0*,
                     3F0*)
```

NOTATION

Each subsection of this manual that describes a control statement starts with a table showing the command and operands for that command, followed by detailed descriptions of the operands. In that table, the following notation is used in this manual to define the usage of each operand:

- Uppercase letters and words must be coded exactly as shown; they are also printed in bold.
- Lowercase non-bold letters and words represent variables that you must substitute specific information as defined in the detailed descriptions.
- When an operand may have several values, they are separated by a vertical bar. You must choose one of the values. For example, PACING=STATIC|DYNAMIC means that you can code PACING=STATIC or PACING=DYNAMIC.
- When a particular value is the default for an operand, it is underlined. For example, PACING=<u>STATIC</u>|DYNAMIC indicates that PACING=STATIC is the default.
- Some operands cannot appear together on a given statement. These and other restrictions are given in the detailed operand descriptions.

TABLE OF CONTENTS

PAGE NO.

FDRPAS™, FDRMOVE, AND FDRERASE VERSION 5.4

OVERVIEW III
CONTROL STATEMENT FORMAT V

SUMMARY OF MODIFICATIONS

SUMMARY OF MODIFICATIONS FOR V5.4 LEVEL 80 XI
SUMMARY OF MODIFICATIONS FOR V5.4 LEVEL 78 XII
SUMMARY OF MODIFICATIONS FOR V5.4 LEVEL 76 XIII
SUMMARY OF MODIFICATIONS FOR V5.4 LEVEL 75 XIII
SUMMARY OF MODIFICATIONS FOR V5.4 LEVEL 74 XV
SUMMARY OF MODIFICATIONS FOR V5.4 LEVEL 73 XV

300 FDRPAS™ INTRODUCTION AND OVERVIEW

300.1 FDRPAS™ CHECKLIST 300-1
300.2 FDRPAS™ OVERVIEW 300-3
300.3 FDRPAS OPERATION 300-6
300.4 FDRPAS ON A SINGLE SYSTEM 300-17
300.5 FDRPAS ON MULTIPLE SYSTEMS 300-18
300.6 FDRPAS HISTORY RECORDS 300-21

305 FDRPAS GENSWAP JCL, STATEMENTS, AND EXAMPLES

305.1 GENSWAP OVERVIEW 305-1
305.2 FDRPAS GENSWAP JOB CONTROL REQUIREMENTS 305-2
305.3 GENSWAP STATEMENT 305-4
305.4 GENSWAP TASK EXCLUDE STATEMENT 305-7
305.5 GENSWAP TASK MOUNT STATEMENT 305-8
305.6 GENSWAP MODEL DEFINITION 305-9
305.7 GENSWAP PROCEDURE AND EXAMPLES 305-14
305.8 FDRPAS FOR LARGE SCALE SYNCHRONIZED MIGRATION 305-30
305.9 GENSWAP CONSIDERATIONS 305-39

PAGE NO.

310 FDRPAS SWAP JCL, STATEMENTS, EXAMPLES, AND ISPF INTERFACE

310.1 FDRPAS JOB CONTROL REQUIREMENTS 310-1
310.2 SWAP, SWAPDUMP, SIMSWAP, SIMSWAPMON STATEMENTS 310-4
310.3 SWAP TASK MOUNT STATEMENT 310-18
310.4 SWAP TASK EXCLUDE STATEMENT 310-19
310.5 MONITOR SWAP STATEMENT 310-21
310.6 MONITOR SWAP MOUNT STATEMENT 310-25
310.7 MONITOR CONFIRM STATEMENT 310-26
310.8 MONITOR CONFIRM MOUNT STATEMENT 310-28
310.9 MONITOR WAITSWAPTERM STATEMENT 310-29
310.10 MONITOR WAITSWAPTERM MOUNT STATEMENT 310-30
310.11 HISTORY STATEMENT 310-31
310.12 HISTORY MOUNT STATEMENT 310-32
310.13 SWAPBUILDIX AND EXPANDVTOC STATEMENT 310-33
310.14 MONITOR RESET AND VARY STATEMENT 310-36
310.15 RESET AND VARY MOUNT STATEMENT 310-37
310.16 LICENSE STATEMENT 310-38
310.20 LICENSE EXAMPLES 310-39
310.21 FDRPAS SWAP TASK EXAMPLES 310-40
310.22 SWAPDUMP EXAMPLES 310-52
310.23 MONITOR SWAP EXAMPLES 310-54
310.24 MONITOR CONFIRM EXAMPLES 310-56
310.25 HISTORY EXAMPLES 310-59
310.26 SWAPBUILDIX AND EXPANDVTOC EXAMPLES 310-60
310.27 VARY EXAMPLES 310-62
310.30 FDRPAS SAMPLE PRINTOUTS 310-63
310.31 FDRPAS ISPF INTERFACE 310-65
310.32 SAMPLE FDRPAS ISPF SESSION 310-81

320 FDRPAS SPECIAL CONSIDERATIONS

320.1 FDRPAS SPECIAL HARDWARE CONSIDERATIONS 320-1
320.2 FDRPAS SPECIAL SOFTWARE CONSIDERATIONS 320-10
320.3 FDRPAS Z/VM AND LINUX FOR SYSTEM Z CONSIDERATIONS 320-20
320.4 FDRPAS E-MAIL NOTIFICATION FACILITY 320-26
320.5 FDR EXTENDED MCS SOFTWARE CONSOLE (FDREMCS) 320-34
320.6 FDRPAS AND IBM GDPS/PPRC HYPERSWAP 320-36
320.7 FDRPAS AND IBM BASIC HYPERSWAP 320-41
320.8 FDRPAS AND EMC AUTOSWAP 320-44

325 FDRMOVE

325.1 FDRMOVE INTRODUCTION 325-1
325.2 FDRMOVE JCL AND STATEMENTS 325-5
325.3 FDRMOVE MOVE FUNCTION WITH EXAMPLES 325-26
325.4 FDRMOVE FASTMOVE FUNCTION WITH EXAMPLES 325-31
325.5 FDRMOVE TARGET VOLUME INITIALIZATION 325-37
325.6 FDRMOVE GENERAL PROCEDURE 325-39
325.7 FDRMOVE CONSOLIDATE VOLUMES WITH MOVE FUNCTION 325-40
325.8 FDRMOVE MOVE ACTIVE DATA SETS WITH FASTMOVE 325-42
325.9 CONSOLIDATE VOLUMES WITH FDRPAS AND FASTMOVE 325-46
325.10 FDRMOVE CONSOLE COMMANDS AND ISPF INTERFACE 325-51
325.11 FDRMOVE SPECIAL CONSIDERATIONS 325-58

PAGE NO.

330 FDRERASE

330.0 INTRODUCTION 330-1
330.1 FDRERASE (ERASING DASD) 330-6
330.2 FDRERASE CHECKLIST 330-8
330.3 FDRERASE JOB CONTROL REQUIREMENTS 330-10
330.4 FDRERASE CONTROL STATEMENTS 330-11
330.5 FDRERASE CONSIDERATIONS 330-25
330.6 FDRERASE EXAMPLES 330-33
330.7 FDRERASE SAMPLE OUTPUTS 330-38
330.8 FDRERASE ISPF INTERFACE 330-41

380 INSTALLATION & MAINTENANCE

380.1 LOADING THE INSTALLATION LIBRARIES 380-1
380.2 ELECTRONIC INSTALLATION 380-2
380.3 CD INSTALLATION 380-8
380.5 INVOKING THE INSTALLATION ISPF DIALOG (PANEL A.I) 380-11
380.13 SET THE FDR GLOBAL OPTIONS TABLE 380-13
380.14 SECURITY OPTIONS (PANEL A.I.4.1) 380-16
380.15 GENERAL OPTIONS (PANEL A.I.4.2) 380-18
380.16 FDRPAS AND FDRERASE OPTIONS (PANEL A.I.4.14) 380-20
380.17 AUTHORIZING PROGRAMS 380-23
380.18 DYNAMIC EXIT INSTALLATION PROCEDURE 380-24
380.19 MAINTENANCE PROCEDURES 380-26

390 MESSAGES AND CODES

390.1 INTRODUCTION TO MESSAGES 390-1
390.2 CONSOLE MESSAGES 390-3
390.3 MESSAGES 390-7
390.4 ABEND CODES 390-53

THIS PAGE INTENTIONALLY LEFT BLANK

SUMMARY OF MODIFICATIONS

SUMMARY OF MODIFICATIONS FOR V5.4 LEVEL 80

SWAPPING LARGE NUMBERS OF VOLUMES

Improvements in the design and better integration with system services allows FDRPAS to be more robust and provide the ability to process a larger number of volumes concurrently, providing substantial savings in time to SWAP volumes. When all the volumes to be swapped are provided, FDRPAS can now create an efficient set of SWAP and MONITOR jobs with the corresponding control cards by utilizing the size of the volume, the controller, and multiple LPARs to get the best throughput.

These improvements provide the ability for FDRPAS to concurrently and non-disruptively migrate upwards of 15,000 volumes in less time, with minimal impact on system resources and ongoing workloads, and reducing the amount of clerical effort needed to create the JCL and control statements needed as well as eliminating the potential for clerical errors. When given the information necessary about local JCL standards, volume serial number ranges of source volumes and target device numbers, FDRPAS dynamically generates SWAP jobs and submits the necessary MONITOR jobs on all the associated LPARs.

These improvements include:

- **Generate JCL and Control Statements** – A new technique (GENSWAP) for generating the control statements for SWAP processes allows a simpler method in migrating a complete SSID or Control Unit. This process generates the necessary control statements for the SIMSWAP, SIMSWAPMON, MONITOR, and SWAP processes. The GENSWAP command sorts all the specified devices by size, control unit, then by SSID and spreads the jobs across the SSIDs to reduce contention. The GENSWAP command creates the JCL and control statements for both the main task and the MONITOR task to ensure they are consistent and in agreement.
- **MONITOR Started Tasks** – Creating MONITOR tasks as Started Tasks (STCs) allows more MONITOR tasks to be run simultaneously removing the limitations imposed by not having enough initiators.
- **SWAP Job Limit Increased to 64 Volumes** – The prior limit of swapping 32 volumes in a single SWAP job has been increased to 64 volumes, if sufficient memory is available below the 16 MB line.
- Substantial reduction in overhead by the SWAP tasks allowing 5000 SWAP tasks running simultaneously on a single LPAR.
- The volsers of target and source volume can be flipped once a SWAP has completed to better work with products such as Global Mirror for z/Series (XRC), GDPS, and others. This is accomplished with the `LABEL=SWAP` operand on the SWAP command.

VOLUME LIMIT INCREASED TO 15,000+ VOLUMES

The prior limit of swapping 10,000+ volumes at the same time has been increased to 15,000+ volumes for FDRPAS SWAP and SWAPDUMP operations. This is especially useful for, but not limited to SWAP and SWAPDUMP processes including: HyperSwap, GDPS, Consistency Groups, and moving a large number of volume at a set point-in-time. This can also be useful for moving an entire data center to a new location.

PERFORMANCE ENHANCEMENTS DURING SWAP FOR PAV

Improvements have been made in the channel programs to reduce I/O wait time during the SWAP operation for PAV volumes. This improves user processing times of very active volumes utilizing PAV while FDRPAS is migrating a volume for SWAP, when `ALLOWPAV=YES` is coded.

QUERY HOST ACCESS SUPPORT

Query Host Access (QHA) is available for the DS8870 in code Release 7.1, GA 6/07/2013, and we expect that QHA will soon become available for the DS8700 and DS8800. We urge all FDRPAS customers to install these microcode levels when available, to support enhanced validation by FDRPAS. Review "Query Host Access Support" in Section 320.1 for more details.

EXCLUDE CPUID= STATEMENT IGNORED

The `EXCLUDE CPUID=` statement is now ignored when the source volume for a SWAP or SWAPDUMP resides on a storage subsystem on which FDRPAS can determine which LPARs have the volume online. This includes most EMC Symmetrix subsystems, and IBM DS8*xxx* subsystems with the Query Host Access feature. See "Query Host Access Support" in Section 320.1 and "Multi-System Determination" in Section 320.1 for more details.

SUMMARY OF MODIFICATIONS FOR V5.4 LEVEL 78

SUPPORT FOR 1TB EAV VOLUMES

Support has been added to support 1 Terabyte (TB) Extended Address Volumes (EAVs) for the FDRPAS, FDRMOVE, and FDRERASE products.

COMMON PAGE DATA SETS

FDRPAS is enhanced to be able to complete a SWAP of a volume containing a common page data set even if the common page data set is updated by a page-out during the SWAP. Formerly, the SWAP failed if a page-out to common occurred, but now FDRPAS recopies the data set on the next pass.

VOLUME LIMIT INCREASED TO 10,000+ VOLUMES

The prior limit of swapping 2000 volumes at the same time has been increased to 10,000+ volumes for FDRPAS operations. This is especially useful for, but not limited to SWAPDUMP processes including: HyperSwap, GDPS, Consistency Groups, and moving a large number of volume at a set point-in-time. This can also be useful for moving an entire data center to a new location. Contact INNOVATION DATA PROCESSING for the *FDRPAS SWAPDUMP for Duplicating a System* document containing additional information for processing a large number of volumes.

FDRPAS EFFICIENCY

FDRPAS and FDRMOVE have been enhanced to use less common storage, reducing CPU overhead, and improving the executing speed.

HYPERSWAP ENABLE AND DISABLE

The process of shutting down and restarting HyperSwap during the time the volumes are being swapped is changed to use ENABLE and DISABLE to reduce the amount of time that HyperSwap is inactive in the system.

FDRMOVE EATTR=OPT

The `EATTR=OPT` operand is added to the FDRMOVE program to allow FDRMOVE to allocate non-VSAM data sets in the Extended Address Space (EAS) of an Extended Address Volume (EAV). FDRMOVE previously supported allocation of VSAM data sets in the Extended Address Space (EAS) of Extended Address Volumes (EAVs).

SUMMARY OF MODIFICATIONS FOR V5.4 LEVEL 76

EXPANDED SUPPORT FOR BASIC HYPERSWAP

Basic HyperSwap is an IBM facility included in z/OS, which automates the swap of a large number of Metro Mirror (PPRC) primary volumes to their secondary devices in the case of planned or unplanned outage.

FDRPAS cannot SWAP a volume while it is being actively managed by HyperSwap. Prior to FDRPAS V 5.4/76, it was necessary to bring down Basic HyperSwap before doing swaps with FDRPAS, and then to bring it back up after the FDRPAS SWAP tasks were complete. HyperSwap would be disabled for a significant period (minutes), leaving the installation exposed in case of a failure. To avoid having multiple periods of exposure, the installation would probably want to do the FDRPAS swaps of all DASD volumes in the HyperSwap configuration at the same time. FDRPAS V 5.4/76 greatly reduces this exposure by using the Block/Unblock facility of HyperSwap (this support requires z/OS 1.12 or above, or z/OS 1.9 through 1.11 with the fix for APAR OA26509). With Block and Unblock, Basic HyperSwap does not need to be brought down to do an FDRPAS SWAP; instead it is just suspended, for a much shorter period. The time depends on the number of DASD volumes being swapped and the number of participating LPARs, and may be only a few seconds. Basic HyperSwap is aware of the FDRPAS SWAP tasks and automatically updates its configuration information. Since the process is so much less disruptive, the installation is able to do the FDRPAS swaps in more convenient groups, such as batches of 32 or 100 volumes at a time.

This support for Basic HyperSwap Block and Unblock is available in V 5.4/76 spin level 4 (available on October 5, 2011).

A new program, **FDRHXBLK**, is added to perform the block and unblock of basic HyperSwap.

MONITOR TASK SERVICE CLASS PERFORMANCE

FDRPAS MONITOR tasks automatically reset themselves to a service class with good performance if they are not already in one. This change alleviates the user's responsibility to ensure that the MONITOR tasks have a high dispatching priority and avoid problems that occur when a monitor does not have a high enough dispatching priority to respond in time to communication from the SWAP job. The new parameter `SRVCLASS=` is used to modify this processing and to specify service classes both for SWAP jobs and MONITOR jobs.

ACTIVE SYSPLEX COUPLING DATA SETS RECOGNIZED

FDRPAS now recognizes a volume containing an active sysplex coupling data set and automatically serializes it in the same way as a JES spool or checkpoint volume.

IXSYSIN DD STATEMENT

FDRPAS dynamically allocates the IXSYSIN data set to VIO instead of DASD to avoid a potential lockout.

CPU TIME REDUCTION

FDRPAS reduces the CPU time used when running many concurrent MONITOR tasks.

SUMMARY OF MODIFICATIONS FOR V5.4 LEVEL 75

Major enhancements have been made to FDRPAS / FDRMOVE product family to make volume Swaps easier and simpler to run. For example, enhanced ISPF panels can now be used to generate all the JCL and control statements required to run FDRPAS functions. Also, a single FDRPAS job can now submit all the necessary MONITOR tasks. Details of all these enhancement are as follows:

SIMSWAPMON PROCESSING

FDRPAS has been enhanced by the addition of a new simulation function called SIMSWAPMON. SIMSWAPMON invokes MONITOR tasks to perform extended checking during a simulated Swap. This checking includes confirming that the necessary MONITOR tasks are responding, that the target volume sizes match the source volume sizes, and that target volumes are offline to all LPARs (if online, the volumes can be varied offline automatically if no allocations exist). SIMSWAPMON can be used before the actual SWAP; when combined with Dynamic Monitoring it can ensure that all required MONITOR tasks that the SWAP needs have been started.

DYNAMIC MONITORING

The usage of the FDRPAS MONITOR tasks has been significantly simplified by the addition of Dynamic Monitoring. In GRS complex and MIM complex environments, SWAP tasks pass to the MONITOR tasks the addresses to be monitored. The MONITOR task on each LPAR dynamically adds these addresses to the list of volumes already being monitored. As a result, MONITOR tasks no longer need MOUNT statements to specify the addresses to monitor.

SWAP PROCESS STREAMLINED TO SUBMIT MONITOR TASKS WHEN NEEDED

In addition to Dynamic Monitoring, FDRPAS simulation and SWAP tasks can now automatically start the required MONITOR tasks. In GRSplex and MIMplex environments, SWAP, SWAPDUMP, and SIMSWAPMON tasks can submit the MONITOR tasks to any LPAR if it detects that no MONITOR task is running there. Automatically starting the required MONITOR tasks removes the need for users to manually submit MONITOR tasks on each LPAR.

FDRPAS ISPF PANELS ENHANCED

The FDRPAS ISPF panels have been enhanced and reorganized to support these new features. In addition, as well as monitoring the status of SWAP tasks, the panels can be used to automatically generate the JCL and control statements for multiple SWAP tasks. Users can easily and quickly generate all the statements needed to swap an entire controller, SSID, volume group, and the rest. For example, specification of the controller serial number generates jobs to Swap all the volumes in that controller.

VTOC AND VVDS VERIFICATION OF SOURCE VOLUME

The integrity of the VTOC and VVDS of each source volume can now be checked automatically during a simulation. When the keyword `CHECKSOURCE=YES` is coded, FDRPAS internally invokes an enhanced version of Compaktor to detect possible error conditions before the volume is swapped. These conditions include duplicate VVDSs on the volume, VVDSs with incorrect volume serial number, and overlapping data set extents.

JES SPOOL AND CHECKPOINT VOLUMES

JES2 and JES3 SPOOL and CHECKPOINT volumes are automatically identified on z/OS 1.7 and higher systems so that they can be managed automatically during the swap process. Previously, customers needed to swap the JES volumes from a FDRPAS SWAP task specified with `MAXTASKS=1` to avoid potential interlocks. With this automatic identification of the JES volumes, SPOOL and CHECKPOINT volumes can be swapped with any number of other volumes or each other.

Since the JES volumes cannot be automatically identified with z/OS systems lower than z/OS 1.7, coding the `JESVOL=YES` operand on the MOUNT statement informs FDRPAS that this volume is a JES SPOOL or CHECKPOINT volume and processes this volume the same as above, or the user can run them with `MAXTASKS=1`.

MOVE TO SMS STORAGE GROUP WITH MORE THAN 255 TARGET VOLUMES

FDRMOVE can now move data sets to more than 255 volumes in an SMS storage group, removing the previous restriction of 255 volumes in a `NEWSTORGRP` or `ENEWSTORGRP` specification. If `STORGRP=` is specified, those volumes are sorted by allocated space with the largest allocation first. If `NEWSTORGRP` or `ENEWSTORGRP` is specified, these volumes are sorted by free space with the largest freespace first. FDRPAS then matches up the target volumes with the source volumes to spread the allocated space evenly throughout the `NEWSTORGRP` and `ENEWSTORGRP` volumes.

MIN#SYSTEMS AND #SYSTEMS OPERANDS REMOVED

Documentation for the `MIN#SYSTEMS` (and its alias `MINNSYSTEMS`) and `#SYSTEMS` (alias `NSYSTEMS`) parameters has been withdrawn. INNOVATION DATA PROCESSING recommends that instead of coding these parameters, users should specify the `EXCLUDE CPUID` statement to identify those LPARs that do not need to participate in Swaps. Contact INNOVATION DATA PROCESSING if you think you need to use the `MIN#SYSTEMS` or `#SYSTEMS` operands. If the source DASD volume to be moved is an IBM RVA, an early Oracle StorageTek SVA, or any subsystem that emulates a 3990-3, contact INNOVATION DATA PROCESSING for documentation on how to migrate this type of DASD with FDRPAS.

SUMMARY OF MODIFICATIONS FOR V5.4 LEVEL 74

GLOBAL OPTION FOR FDRW68 MESSAGES

New option `NONRESPONDING` added to the FDR Global Options Table to specify the default option to be taken for the response to the FDRW68 message. The settings can be DEFAULT, FAIL, or RETRY.

MAXTASKS DEFAULT INCREASED

`MAXTASKS=` default value for the MOVE command of FDRMOVE has been increased from 5 to 8.

SUMMARY OF MODIFICATIONS FOR V5.4 LEVEL 73

PAV SUPPORT

FDRPAS has enhanced its support for Parallel Access Volume (PAV) aliases.

DYNAMIC ALLOCATION EXIT SUPPORT FOR FDRPAS

The FDRMOVE Dynamic Allocation exit has been enhanced to support FDRPAS. With this exit running, dynamic allocations are monitored for requests for data sets that are actively being moved by FDRPAS. The jobs requesting those data sets wait until they are available, instead of failing

FDRMOVE SMS CHANGES

When the data sets being moved by FDRMOVE are SMS-managed, the user can now change their storage and management classes.

THIS PAGE INTENTIONALLY LEFT BLANK

300 FDRPAS™ INTRODUCTION AND OVERVIEW

300.1 FDRPAS™ CHECKLIST

This is a simple checklist of steps to swap one or more DASD volumes to new DASD hardware devices. It is placed first in this manual so that you can more easily find it. Please review the rest of the manual to understand the steps in this checklist.

BEFORE YOUR FIRST SWAP

- ❒ Important: Review Section 320.1 "FDRPAS Special Hardware Considerations" and Section 320.2 "FDRPAS Special Software Considerations" for special hardware and software considerations that may affect your swap.
- ❒ Visit the INNOVATION DATA PROCESSING Customer FTP Access web site at http://fdr.com/ftp/ftp.cfm to obtain the latest fixes and updates for FDRPAS, and recommended maintenance from IBM and other vendors. See member "FDRPAS_IBM_and_ThirdParty_Maintenance_*mm-yyyy*.pdf" found in the **Please_ReadMe_First** directory for the latest information.
- ❒ Run the FDRPAS EPSPT job (member "FDRPAS-EPSPT-JOB.txt") found in the **Public/Download/FDRPAS/Please_ReadMe_First** directory. This job uses IBM's Enhanced Preventive Service Planning Tool (EPSPT) to automate checking your system for APARs that need to be applied to all of your systems before you attempt to use FDRPAS. INNOVATION DATA PROCESSING strongly recommends that you use EPSPT rather than manually checking all the APARs.
- ❒ Install FDRPAS as shown in Chapter 380 "Installation & Maintenance" and APF authorize the installed load library on all participating LPARs. Be sure to install the PASPROC cataloged procedure in a system procedure library and assign the FDRPAS catalog alias to a user catalog (optional) so that FDRPAS can record history records. Install the FDRPAS ISPF dialogs (see Section 380.5 "Invoking the Installation ISPF Dialog (Panel A.I)"). The ISPF panels should be used to create and submit the SIMSWAP, SIMSWAPMON, SWAP, and MONITOR tasks.
- ❒ Log on to the INNOVATION DATA PROCESSING FTP site at http://fdr.com/ftp/ftp.cfm and traverse to the **Maintenance** directory to check that your site is up-to-date on FDRPAS maintenance.
- ❒ Run the `LICENSE TYPE=SWAP` job (see "License Report Example" in Section 310.20) and forward the output to your INNOVATION DATA PROCESSING sales representative or e-mail it to sales@fdrinnovation.com.
- ❒ Please complete the Hardware/Software Profile at the end of this manual, and fax it to your INNOVATION DATA PROCESSING representative. Alternatively, you can e-mail it to support@fdrinnovation.com.

SWAPPING ONE OR MORE VOLUMES

- ❒ Instruct your system console operators not to reply to any FDRPAS console messages without your knowledge. Do not automate the replies to any FDRPAS console messages. You can specify the operand `WTOR=NO` to prevent the operators from replying to FDRPAS messages inappropriately.
- ❒ Ensure that the target device is defined in the configuration of every system image that uses the source volume to be swapped. The target device must be varied offline on the system where the SWAP task is going to run, but it **must not** be marked as offline in the I/O configuration. If the target device is in newly installed hardware, you can use the dynamic I/O configuration function of HCD to add it to the configuration. Target devices that are not offline on the systems where the MONITOR tasks run are varied offline by the MONITOR task if no allocations exist on the volume.
- ❒ If the source volume is currently in a DASD subsystem such as IBM RVA, early Oracle StorageTek SVA, or any that emulate an IBM 3990-3 control unit, contact INNOVATION DATA PROCESSING on how to move these subsystems.

- ❒ If the source volume is currently in a DASD subsystem such as an IBM 3990-6, IBM 2105, IBM 2107, DS6000, DS8000, EMC, HDS or any that emulate an IBM 3990-6, 2105, or 2107 control unit, FDRPAS can determine which LPARs the DASD is connected to and reports their CPU serial number.
- ❒ Now you are ready to start using the FDRPAS ISPF dialogs to create a list of source volumes and assign target addresses. You can go to Section 310.31 "FDRPAS ISPF Interface" to see a sample of all the panels, jobs, and commands that are available. After entering source and target information, the status field shows possible problems including:
  - ❍ Target unit does not exist
  - ❍ Target device not offline
  - ❍ Source and target size mismatch.
- ❒ Before generating any jobs, you should use the Options dialog to set the FDRPAS STEPLIB data set name, CPUID and JCL requirements for the MONITOR tasks, as well as setting defaults for the SIMSWAP, SIMSWAPMON, and SWAP jobs (for example, `CHECKTARGET=YES` and `CHECKSOURCE=YES`).
- ❒ Create a SIMSWAP job by using the ISPF panels. The `CHECKTARGET=YES` checks that the target volumes are offline and empty. `CHECKSOURCE=YES` examines the VTOC and VVDS of the source volumes to determine if they contain errors, such as:
  - ❍ Active PAGE and SWAP Data Sets on source – FDRPAS can not move. Suggest you define new page and swap data sets on new volumes (see "PAGE and SWAP Data Sets" in Section 320.2)
  - ❍ Overlapping extents – FDRPAS moves them, but the overlapping extents still exist
  - ❍ VTOC volume size does not match the device size – FDRPAS can not move. Review message FDR124 REASON=4 for more information
  - ❍ VVDS does not match – FDRPAS moves them, but the invalid VVDS still exists
  - ❍ Duplicate VVDSs on the volume – Contact INNOVATION DATA PROCESSING regarding movement of a volume with this situation.

  In addition, the SIMSWAP job provides you with a list of CPU serial numbers of the LPARs that are connected to the source volume(s) specified in the MOUNT statement(s).
- ❒ Create a FDRPAS SIMSWAPMON job by using the FDRPAS ISPF dialogs to examine all of the volumes to be monitored. This also checks the specified devices on all the other systems in the PLEX to ensure that they are added to the list of watched devices. Use the same parameters on the SIMSWAPMON that are used in the actual SWAP job.
- ❒ If z/VM systems are involved in the SWAP process, please see Chapter 321 "FDRPASVM" for considerations by using the FDRPAS monitor for z/VM systems and start the FDRPASVM monitor for the same devices specified on the SIMSWAPMON job.
- ❒ As soon as the SIMSWAPMON is fine, you are ready to create the actual SWAP job(s). You are now sure that all common conditions that make a swap fail have been verified before the actual swap is started. In the event that the actual swap is run much later than the SIMSWAPMON, it is recommended to repeat the SIMSWAPMON job to ensure that everything is still in place.
- ❒ If you use FDRPAS with FDRMOVE, please see Section 325.11 "FDRMOVE Special Considerations".

RECOMMENDATION: ***To all FDRPAS users prior to V5.4L75.*** *In many cases, you used the FDRPAS ISPF dialogs to only monitor the swap jobs. With FDRPAS V54L75 or higher, you should now use the FDRPAS ISPF dialogs to create all of jobs including the SIMSWAP and SIMSWAPMON jobs prior to the migration. This makes the migration effort much easier and error free.*

300.2 FDRPAS™ OVERVIEW

FDRPAS™ (FDR Plug and Swap™) is used to move z/OS DASD volumes non-disruptively from one DASD volume to another, and to create non-disruptive point-in-time backups of DASD volumes.

FDRPAS allows a z/OS installation to:

- Swap DASD volumes from their current locations to new DASD hardware.
- Move DASD volumes within the installation for load balancing.
- Create point-in-time backups of DASD volumes.
- Create duplex copies of DASD volumes.
- Do this volume movement without interrupting any system activity.
- Do this volume movement during normal system operations.
- Swap shared DASD on all sharing systems simultaneously.
- Swap many DASD volumes concurrently.
- Monitor and control FDRPAS functions with ISPF panels.

FDRPAS BENEFITS

- The operating system, application jobs, online systems, and users are unaware that FDRPAS is swapping DASD volumes to new DASD devices.
- A 24x7 installation, with no window for major re-configurations and hardware changes, can install and activate new DASD hardware.
- Even installations that do have dedicated maintenance windows may choose to use FDRPAS to swap to new DASD hardware during normal business hours.
- No matter how big your maintenance window is, it is not big enough to move the terabytes of data in most installations. FDRPAS removes that limitation.
- FDRPAS can be used for load balancing within your existing DASD hardware.
- FDRPAS can create point-in-time backups without special hardware features.
- It can create duplex copies of volumes, between control units from different vendors, without special hardware connections. These copies can be in local or remote subsystems.

SUPPORTED SOFTWARE & HARDWARE

FDRPAS executes on any release of z/OS.

FDRPAS supports a wide variety of DASD devices from hardware vendors including IBM, EMC, Oracle StorageTek, and Hitachi and HP. It can swap DASD volumes between DASD of the same type from the same hardware vendor or different hardware vendors without any special software or hardware modifications.

FDRPAS supports DASD volumes attached via FICON, ESCON, and parallel (bus/tag) channels. FICON and ESCON directors and channel extenders are supported.

SUPPORTED VOLUME TYPES

FDRPAS can swap any z/OS volume including the SYSRES volume, other system volumes, open catalog volumes (user and master catalogs), application data volumes, CICS volumes, database volumes, Hierarchical File System (HFS) and zSeries File System (zFS) volumes, LINKLIST volumes, TSO volumes, SMS-managed volumes, and work volumes. The only exceptions are volumes containing active local PAGE or SWAP data sets (see "PAGE and SWAP Data Sets" in Section 320.2).

MOVEMENT TO NEW HARDWARE

z/OS hardware and software allows you to attach new DASD subsystems (hardware) to your system and dynamically activate an updated I/O configuration to make them available. FDRPAS complements that capability by allowing you to move your DASD volumes to this new hardware while those volumes are still in use.

When the FDRPAS swap of a volume is complete, the volume resides completely on the new device, and the original device is no longer required. If all the volumes on an old DASD subsystem are moved to new locations with FDRPAS, the old subsystem can be powered off and disconnected.

Without the use of FDRPAS, implementation of new DASD subsystems might require that many of your applications, perhaps your entire system, be shut down while volumes are backed up and restored to new locations. The conversion process may take many hours, even days, and often needs to be done during evenings or weekends.

FDRPAS can be used to move production application volumes to new hardware, to see if the hardware meets performance expectations or claims. If it does not, FDRPAS can easily move the data back to its original location.

With FDRPAS, the new hardware can be implemented at any time, even during prime time, without interrupting any of your normal workload or activities. FDR Plug and Swap truly allows you to "plug" in new DASD hardware and "swap" your DASD data to the new DASD non-disruptively.

I/O LOAD BALANCING

FDRPAS can also be used to move DASD volumes for load balancing purposes while those volumes are still in use.

System performance monitoring may discover that certain channel paths, DASD subsystems, or sets of DASD hardware are overloaded because of the current placement of certain data sets or databases. These overloads may cause your service times to degrade, and the problem can get worse over time.

The normal response to such overloads is to live with the problem until time can be scheduled to shut down the affected applications and move volumes or data sets until the I/O loads are more evenly balanced.

With FDRPAS, volumes can be moved within your system to re-balance your I/O loads at any time, while the applications are running. I/O load balancing can become a regular part of your system performance tuning.

POINT-IN-TIME BACKUPS

FDRPAS can also be used to create non-disruptive point-in-time backups of DASD volumes, when used in conjunction with FDRINSTANT, a feature of the FDR family of DASD management software products.

In this mode, FDRPAS copies an online DASD volume to an offline target, and then optionally continues running to keep the offline copy in synchronization with the online DASD volume. At the point when you want to take the backup of the online DASD volume, you terminate FDRPAS and the offline DASD volume contains an exact point-in-time copy of the online data. FDRINSTANT allows you to back up that offline copy as if you were backing up the online DASD volume, except that the data is frozen at that point-in-time.

FDRINSTANT also works with various hardware functions that allow for the creation of point-in-time copies, such as SnapShot, FlashCopy, TimeFinder, and ShadowImage. FDRPAS can be used with FDRINSTANT when none of those hardware functions are available; it supports FDR and FDRDSF backups and FDRCOPY, but not FDRABR at this time.

DUPLEX COPIES

FDRPAS can also be used to create duplex copies of volumes, without requiring special hardware connections. These duplex copies can be in subsystems that are locally attached or remotely attached over extended-distance connections or channel extenders. There are no hardware restrictions; the duplex copy of a volume can be in any DASD subsystem, even if it is from a different vendor than the primary copy.

These duplex copies are updated asynchronously. FDRPAS copies each volume and then monitors the primary volume for updates, re-copying updated tracks as necessary to keep the volumes closely synchronized.

At any point, the duplex volume can be split from the primary and can be used as a copy of the primary volume.

INDEXED VTOC MAINTENANCE UTILITY

FDRPAS includes a utility function to create an Indexed VTOC (VTOCIX) or rebuild a disabled VTOCIX on a volume even while that volume is online and in use on multiple systems. This is similar to the BUILDIX function of the IBM ICKDSF utility. The FDRPAS SWAPBUILDIX function uses the cross-system coordination functions built into FDRPAS to allow the BUILDIX to be recognized on all systems without taking it offline.

300.3 FDRPAS OPERATION

FDRPAS can swap volumes in use on a single system image, as well as those attached to multiple systems or LPARs in a shared DASD complex or sysplex, whether locally or remotely attached. Multiple volumes can be swapped concurrently.

FDRPAS TASKS

FDRPAS operates as two kinds of tasks:

1. The active **SWAP** task. This task initiates the swap of one or more DASD volumes to new DASD devices. It copies the data tracks from the source volume to the target DASD, and causes the operating system to swap all I/O to the target when the DASD volumes are synchronized. A single SWAP task can swap up to 64 DASD volumes concurrently. If you need to swap more than 64 volumes concurrently, you must start multiple SWAP tasks.
2. The update **MONITOR** task. This task monitors one or more offline potential target DASD devices. It detects that a swap has begun on a DASD volume and installs I/O intercepts that monitor all I/O to the source volume for updates. It also causes the operating system to swap all I/O to the target when the DASD volumes are synchronized. You can use one MONITOR task (per system image) to monitor all potential target devices or you may choose to start multiple MONITOR tasks on each system image with each task monitoring a set or range of targets. You can even start one MONITOR task per target, if you prefer. If a MONITOR task is monitoring multiple target devices, it actually starts additional MONITOR tasks (one per target device) as internal subtasks or external started tasks when a swap request is detected.

When multiple CPUs or LPARs (“system images”) have access to a volume to be moved, the SWAP task for each volume executes on only one system image, but the MONITOR task must execute on **all** system images with access to the volume (up to 128 system images are supported). The SWAP task also acts as the MONITOR task on the system that it is executing.

These tasks can be executed as submitted batch jobs, or as started tasks executed on-demand, although we refer to them as “tasks” in this manual.

NOTE: Although FDRPAS SWAP and MONITOR tasks use very little CPU time, they must be able to get access to the CPU when they need it in order to maintain the communication between systems. INNOVATION DATA PROCESSING recommends that you run all FDRPAS tasks at a high dispatching priority, such as by putting them in a WLM class with a high velocity.

FDRPAS VOLUME SWAP

Swap of a DASD volume is very simple. An FDRPAS MONITOR task is started on each system that has access to the target device, monitoring that device. On one system, an FDRPAS SWAP task is started to initiate the swap of the online source volume to the offline target DASD device. It is usually desirable to execute the SWAP task on the system with the most update activity on the volume; however, if you are executing many swaps concurrently, you should spread the SWAP tasks across as many systems as possible.

The FDRPAS SWAP task communicates with the MONITOR tasks on all other systems to coordinate the swap operation. It verifies that every system that can see both the source and target volumes is involved in the swap. FDRPAS starts the swap only if the target device is offline to all sharing systems where the source volume is online to ensure that an active volume cannot be accidentally overlaid. **However, FDRPAS cannot detect a target volume that is online to a system where the source volume is offline, so you must ensure that the target volume is not in use anywhere.**

The FDRPAS SWAP task copies all allocated tracks (for some data sets, only used tracks) on the source volume to the target volume, while simultaneously detecting all updates to the source volume; updated tracks are re-copied if necessary so that the target volume eventually contains an exact image of all of the active data on the source volume. The target volume remains offline to z/OS during the copy, so that the copied data is protected until the swap is complete.

Once the copy is complete and the two devices are completely synchronized, FDRPAS completes the swap by asking the operating system to re-direct all I/O for the volume from the original source volume to the new target device on every system involved. The new device effectively replaces the original, and the original DASD volume is placed offline. All existing jobs, tasks, and users that were allocated to the volume are now allocated to the target device, although they are unaware that the swap has taken place.

When the swap is complete, the volume label on the old source volume is modified so that the operating system is no longer able to vary it online. When the system is next re-IPL'd, it finds the volume on the target device and does not attempt to use the old source volume. To be sure that this occurs, do **not** mark the target devices offline in your I/O configuration.

Once all volumes in a DASD subsystem have been swapped to new devices, you can power off and disconnect the old subsystem, if that is your intention. If you want to reuse the old device for some purpose, you can do an offline INIT with the IBM ICKDSF utility (specifying NOVERIFY) to give it a new volume serial, or you can execute the FDRPAS MONITOR TYPE=VARYONLINE function (see Section 310.14 "MONITOR RESET and VARY Statement") to modify the volume label on the original device so that it can be re-mounted, if you need to do this.

Only the source and target devices are accessed by FDRPAS during the swap. It does not use any additional communication between systems. It **does not require** TCP/IP, VTAM, a data set on a third DASD volume, or a coupling facility.

The swap is accomplished with minimal impact on the performance of applications using the volumes being swapped. Applications continue to execute, unaware that the data movement is occurring or has completed. FDRPAS manages the copy to minimize its effect on the system. For example, inactive data sets are copied first, and tracks within active data sets that are updated are deferred until the end of the copy, so that they do not have to be copied many times. If the FDRPAS copy I/O is noticeably impacting system performance, you can request that the FDRPAS I/O be paced, adding a small delay between each I/O to allow other applications access to the DASD volumes and channels; I/O pacing can be dynamically modified during the swap process.

Swapping of a volume can be terminated at any time before the final swap without affecting the original device or any applications using it. FDRPAS ISPF panels can be used to terminate the swap. Alternately, you can cancel a SWAP task and all of the active swaps in that task terminate with an error.

Operating system swap services are invoked to perform the final swap. As a result of this swap service, the Unit Control Block (UCB) of the source and target volumes are swapped

in memory, so that the original source UCB now points to the new device, and vice versa. This allows the UCB pointers of all jobs, tasks, and users who have the source volume allocated to remain unchanged and unaware that a new device is in use. Note that if your installation has JCL or dynamic allocation that allocates by using the actual unit address (for example, UNIT=3FA), they need to be changed after the swap.

After a successful swap, the now-offline original device can be used as a point-in-time backup of the volume, at the point of the final swap. If you are using FDRPAS to migrate to new hardware, when all volumes in the old DASD subsystem have been swapped to new DASD volumes, the old subsystem can be disconnected and removed.

SWAP PHASES

The operation of FDRPAS is divided into five (5) phases:

1. Phase 1: Initialization

This phase begins when a swap is requested by an FDRPAS SWAP task as well as during the SIMSWAPMON process. The swap request is validated and, if multiple systems are involved, the FDRPAS MONITOR tasks on the other systems are notified of the swap request. Since the SIMSWAPMON task performs the same processing as the swap process, it is highly recommended to run this to ensure that the swap process runs without errors. The SIMSWAPMON task prepares for the real swap and ensures a cleaner swap process without performing the swap.

- If `CONFMESS=YES` was specified, FDRPAS asks the system operator for permission to continue, via a WTOR with message FDRW01. You can also reply to this message from the FDRPAS ISPF panels. If `WTOR=NO` is specified, the console message is only a WTO and you **must** use the ISPF panels to reply.
- FDRPAS verifies that the specified source volume and target device are valid for a swap, making sure that they are the same DASD device type, that the target is offline, and that the source is eligible to be swapped. It also checks if the devices have the same number of data cylinders unless `LARGERSIZE=OK` is specified; in that case the target can be larger. If FDRPAS security is enabled, FDRPAS verifies that the security user id associated with the SWAP task has proper authority.
- If multiple systems have access to the source volume, the SWAP task indicates that a swap is beginning and waits for the MONITOR tasks on the other systems to acknowledge that they are ready to participate. On the FDRPAS ISPF panels, the status shows as SYNCHRONIZING.
- Each MONITOR task acknowledges that it has access to both the source and target devices, that the target is offline, and that they are ready to participate. If the target device is not offline on an LPAR that a MONITOR task is running, special checking is done by the MONITOR task to ensure that this device is the same target device as specified by the main FDRPAS process and that the device is inactive on the LPAR this MONITOR task is running on. If so, then this volume is varied offline on by this MONITOR task. If a system can access the target device but not the source volume, the MONITOR task indicates that it does not need to participate.
- When the proper number of MONITOR tasks have acknowledged that they are ready to participate, the SWAP task proceeds. If the expected number of systems have not acknowledged within a time limit, this probably means that an FDRPAS MONITOR task for the target device was not running on all required systems, that the target was not offline on one or more systems, or that one or more systems does not have access to the target. You must run a MONITOR task on every system that has access to the source volume, even if it is offline, and those systems must also have access to the target device. FDRPAS asks if it should continue with the swap even though the expected number of systems are not participating by issuing message FDRW68 (unless you specify `NONRESPONDING=FAIL`). You should not reply “YES”. Contact INNOVATION DATA PROCESSING if you are not able to resolve this with a “RETRY”.
- Note that the FDRW68 message is issued as a WTOR, to which the system operator can reply, by default. You can also display and reply to the message by using the FDRPAS ISPF panels. Optionally, you can change the message to a simple WTO so that the system operator cannot reply; in this case the ISPF panels must be used to reply. If the system operator is not involved in the swaps, the `WTOR=NO` operand is recommended to prevent erroneous replies.

2. Phase 2: Activation

The SWAP task signals that Phase 2 has begun. On each system, FDRPAS temporarily suspends all application and system I/O to the source volume and install an I/O intercept to monitor updates to the source volume. When this is done on all systems, I/O is allowed to proceed. The swap has now begun. The time required to complete Phase 2 varies depending on the number of systems involved. On the FDRPAS ISPF panels, the status now shows as ACTIVE.

3. Phase 3: Copy

The SWAP task copies data tracks from the source volume to the target device, reading and writing up to 15 tracks per I/O.

- The first pass of the Phase 3 copy copies all tracks on the source volume. Only tracks currently allocated to a data set are copied, plus tracks in the VTOC, VTOC index, VVDS and volume label. For Physical Sequential (PS), Partitioned Organization (PO), and VSAM data sets, only used tracks are copied unless those data sets are allocated to some job or task at the beginning of the swap, in that case all allocated tracks are copied.
- While the Phase 3 copy is progressing, the I/O intercepts on each system are monitoring I/Os to the source volume to identify tracks that are updated. At the end of each pass of Phase 3, a consolidated list of updated tracks is collected (see Phase 4) and an additional pass of Phase 3 is made to re-copy those updated tracks. These additional Phase 3 passes continue until the number of tracks remaining to be copied is small.
- Before a track is copied, FDRPAS checks to see if the I/O intercept on the system running the SWAP task has determined that the track was updated during the current pass, and defers copying the track until the next pass. This avoids unnecessarily copying tracks that just need to be re-copied.

4. Phase 4: Update Consolidation

At the end of each Phase 3 copy pass, Phase 4 is entered and the SWAP task requests a list of updated tracks from each MONITOR task. I/O to the source volume is suspended briefly on all systems while this information is collected. A consolidated list of tracks updated on all systems is formed. FDRPAS determines if it can complete the swap:

- If the number of tracks in the list is above a threshold, Phase 3 is re-entered to re-copy the updated tracks. Note that after every Phase 3 pass, the threshold value is increased, in case the rate of updates to the source volume is very high.
- If the number of tracks in the list is below the threshold or there are no updated tracks in the list, then FDRPAS is ready to complete the swap.
- If `CONFIRMSWAP=YES` was specified on the SWAP statement, then you do not want the swap to complete until you tell it to, so FDRPAS simply re-enters Phase 3 to copy the updated tracks (INNOVATION DATA PROCESSING does not recommend the use of `CONFIRMSWAP=YES` unless you need to complete the swap of many volumes at the same time). This continues until you confirm the swap (if the number of updated tracks again rises above the threshold, the volume no longer is “ready to swap” until it falls again). If there are no tracks in the update list, FDRPAS simply waits for an interval and test for updates again. You can confirm the swap in two ways: the FDRPAS ISPF panels can be used to monitor the progress of the swaps and confirm the swap of one or more volumes, or you can submit a `MONITOR TYPE=CONFIRMSWAP` job to wait for one or more DASD volumes to become ready for completion and automatically confirm the swap. `CONFIRMSWAP=YES` does not result in any console message or WTOR.
- If `CONFIRMSWAP=NO` was specified or defaulted, then FDRPAS automatically completes the swap as soon as the number of updated tracks in Phase 4 falls below the current threshold.
- On every system, FDRPAS disables all application and system I/O to the source volume, then enters Phase 3 for one last pass to copy the remaining updated tracks (unless the updated track list is empty). Depending on the current value of the threshold and the number of tracks in the list, I/O is suspended from as little as a few seconds to, in the worst case, a minute or more. This quiesce time depends on the number of participating systems and the number of updated tracks to be copied.

5. Phase 5: Swap Completion

At this point the source and target devices are completely synchronized. On every system, FDRPAS invokes operating system services to swap the devices. The volume now appears to be mounted on the target device that is now online, all future I/O is directed to the target device, and all jobs, tasks and users that have the volume allocated are now pointed to the target device. The original source volume is placed offline and its volume label is modified so that it cannot be accidentally placed online again. FDRPAS removes its I/O intercepts on all systems and re-enable I/O to the volume. The swap is complete. On the FDRPAS ISPF panels, the status shows as COMPLETED but only for swaps that previously had a status of ACTIVE.

AUTOMATIC SWAP TERMINATION

If the MONITOR task on any system fails to respond in any phase of the swap (except Phase 5), the SWAP task automatically terminates the swap. This probably means that a MONITOR task has abnormally terminated or been canceled, or a system involved in the swap has crashed or been shut down.

Similarly, if the SWAP task is abnormally terminated or canceled, or the system executing the SWAP task crashes or is shut down, the swap is terminated.

If an I/O is issued to the source volume on any system that contains Channel Command Words (CCWs) that are not recognized by FDRPAS, the swap is terminated, since FDRPAS cannot tell if that I/O has updated the source volume, or what tracks it has updated. This probably means that the source volume DASD subsystem supports special vendor-specific CCWs for functions that are unknown to FDRPAS. In this case, FDRPAS prints some diagnostic information about the suspect CCW chain and the job that issued it. You should contact INNOVATION DATA PROCESSING with this printout so that we can attempt to identify the CCWs and enhance FDRPAS to handle them properly. If you can determine that the job has used functions that are restricted during an FDRPAS operation (such as Concurrent Copy (CC), see Chapter 320 "FDRPAS Special Considerations"), you may be able to re-execute FDRPAS at a time when those functions are not in use.

WARNING: ***If a system with access to the source volume is IPL'd (activated) or an offline source volume is varied online on some system while a swap is in progress for that volume, FDRPAS is not invoked on that system and it does not participate in the swap. FDRPAS may be unable to tell that this has occurred, so when it enters Phase 5, the volume is not swapped on this new system and updates to that volume on that system are done on the wrong device; if FDRPAS can detect that the IPL or VARYON has occurred, the swap is terminated in Phase 5. You should avoid IPLing systems during FDRPAS swaps unless they do not have access to the source volumes involved. You should not VARY volumes involved in swaps online.***

ELIGIBLE VOLUMES FOR SWAP

All volumes are eligible to be swapped except for those containing active local page or swap data sets. These volumes can be moved by creating and activating new page data sets on other volumes and deactivating those on the volumes to be swapped.

The system residence (IPL) volume can be swapped, but you must be sure to update your IPL parameters on all affected systems with the new IPL address before the next IPL.

However, you should read Chapter 320 "FDRPAS Special Considerations" carefully; since there may be steps you need to take before moving certain volumes.

POINT-IN-TIME BACKUPS

When FDRPAS is used to create a point-in-time backup (the SWAPDUMP statement), the operation of FDRPAS is similar to the operation of a normal swap except that the volumes are not swapped at the end of the operation. FDRPAS simply terminates, leaving the target device with an exact copy of the source volume (except that the label is changed from "VOL1" to "FDR1") at the point that FDRPAS ended.

Start an FDRPAS SWAPDUMP operation for all volumes involved in the backup well before the backup is to be taken to give FDRPAS time to synchronize all those volumes. Volumes involved in a SWAPDUMP backup cannot also be involved in a true swap, and no more than one SWAPDUMP can be in operation for a given volume at one time.

Normally, you want to specify the `CONFIRMSPLIT=YES` operand on the SWAPDUMP statement. This operates identically to the `CONFIRMSWAP=YES` operand of the SWAP statement, causing FDRPAS to continue to operate even when the volumes are synchronized, recopying updated tracks as necessary to maintain the synchronization. You must "confirm" the volumes through the FDRPAS ISPF interface or by submitting a `MONITOR TYPE=CONFIRMSPLIT` statement that terminates FDRPAS and make the offline target volumes available for dumping when you are ready to take the backup of the volumes.

FDRPAS SWAPDUMP supports FDRINSTANT backups with FDR and FDRDSF, and data set copies with FDRCOPY. It does not support FDRABR® backups at this time.

CONFIRMSWAP AND CONFIRMSPLIT

By default, a SWAP operation (to actually move a volume) and a SWAPDUMP operation (to create a point-in-time backup) complete automatically as soon as the source volume and target device are synchronized or when only a small number of data tracks remain to be synchronized. No operator or user intervention is required to complete the operation.

However, the CONFIRMSWAP=YES operand (for SWAP) and CONFIRMSPLIT=YES operand (for SWAPDUMP) can be used to allow the operator or user to control when the operation on a given DASD volume completes. If these operands are specified, then FDRPAS enters an "idle" state when the devices are synchronized or close to synchronization. In this state, FDRPAS continues monitoring the source volume for updates and re-entering Phase 3 (as documented earlier) to periodically copy the updated tracks, to keep the devices in close synchronization. However, it continues to do this indefinitely until it is instructed to complete the operation.

Why would you want to do this? For a SWAP, you generally do not want to use `CONFIRMSWAP=YES` unless you have some special reason for wanting to control when the swap to the new device actually occurs. When swapping a single volume, there is rarely any reason to do so, since you usually want the swap to complete as soon as possible. Even when swapping many volumes in parallel, you usually want to let each volume swap as soon as it is synchronized. However, if you have some reason that you need to co-ordinate the actual swaps, you can use `CONFIRMSWAP=YES`. **In most cases, you should omit `CONFIRMSWAP=YES`.**

For a SWAPDUMP, `CONFIRMSPLIT=YES` may make sense, since it allows you to control the time that the point-in-time backup is frozen. It may be especially useful when creating point-in-time backups of many DASD volumes, so that they can all be frozen at approximately the same time.

`CONFIRMSWAP=YES` and `CONFIRMSPLIT=YES` do not result in any console messages or WTORs (although some users seem to expect that they do). There are two ways to tell FDRPAS to complete the operation:

1. If you use the FDRPAS ISPF panels to monitor FDRPAS operations, the panels tell you which SWAP and SWAPDUMP tasks have used the confirm operand, and also tell you when each volume has reached synchronization and is ready to confirm. You can then enter a command on the panel to confirm one or more DASD volumes and complete their operations.
2. If you want to automate the process, you can use an FDRPAS job or started task with the `MONITOR TYPE=CONFIRMSWAP` or `TYPE=CONFIRMSPLIT` statement (Section 310.7 "MONITOR CONFIRM Statement"). This is followed by one or more MOUNT statements (Section 310.8 "MONITOR CONFIRM MOUNT Statement") identifying DASD volumes. When all of the volumes identified are in the "ready to confirm" state, they are all confirmed automatically. This is an easy way to automatically complete the SWAP or SWAPDUMP operation for a set of volumes at the same time.

I/O PACING

By default, FDRPAS does I/O to the source and target devices as rapidly as the hardware and operating system allow. Up to 15 tracks are read or written per I/O (unless overridden by BUFNO=). This allows FDRPAS to complete the swap of a volume very quickly. The swap of a 3390-3 typically completes in two to three minutes, depending on the number of tracks to be copied, source and target device types, and so on.

If there is I/O activity on the volume from other applications or the system, the FDRPAS I/O may have an impact, causing the other I/O to be delayed or elongated. In most cases, this degradation is not noticeable; batch jobs that are using the volume may run a little longer and online users may see a slight increase in response time. Since the degradation vanishes as soon as the swap is complete, there is usually no need to be concerned about it. If you are swapping volumes to newer, faster hardware, response time improves as soon as the swap is complete, so it is desirable to complete it as quickly as possible.

However, you may have an environment where online response time or batch service times are extremely important so that the FDRPAS degradation is not acceptable. The obvious solution is to run FDRPAS off-hours when the impact is not noticeable, but if that is not practical, FDRPAS includes I/O pacing options to reduce the impact of its I/O.

FDRPAS I/O pacing works by inserting a time delay between WRITE I/Os to the target device. This also causes delays between READ I/Os on the source volume (note that if the target hardware is significantly faster than the source, it may require large pacing delays before the source I/O is delayed).

Static I/O Pacing – is invoked by specifying the PACEDELAY=*nn* operand on the SWAP or SWAPDUMP statement. This introduces a fixed delay of *nn* hundredths of a second between writes. The PACEDELAY= value can also be interactively modified from the FDRPAS ISPF panels, even if it was not specified when the swap was started. Therefore, if the FDRPAS I/Os are causing unacceptable degradation, you can change the pacing values up and down from the panels until you are satisfied with the results.

Dynamic I/O Pacing – is invoked by specifying PACING=DYNAMIC on the SWAP or SWAPDUMP statement. When in use, FDRPAS uses an algorithm to gauge the impact of the FDRPAS I/Os on queue lengths and I/O delays on the source volume. Every 15 seconds, it may increase or decrease the PACEDELAY= value in use (from 0 to 50), depending on recent results. If you also specify the PACEDELAY=*nn* operand, it is used as the initial pacing value; otherwise the initial value is determined by FDRPAS when the swap starts (the maximum initial value is 20). You can observe the pacing value from the ISPF panels, and you can change it if desired (FDRPAS starts adjusting the pacing from the new value).

NOTE: I/O pacing, either static or dynamic, causes the swaps to take longer. In most cases, it is better to complete the swap as quickly as possible without using pacing. **INNOVATION DATA PROCESSING recommends that you do not use static or dynamic I/O pacing unless you have experienced unacceptable degradation due to the use of FDRPAS.**

TERMINATING FDRPAS

FDRPAS SWAP tasks terminate automatically when all volumes requested by MOUNT statements have been processed (successfully or unsuccessfully).

FDRPAS MONITOR tasks with DYNMON=NO terminate automatically when all target devices being monitored (as specified on MOUNT statements or added dynamically) have been successfully swapped. It determines this by checking if the target devices are now online, so varying them online also terminates the MONITOR task. However, if the MONITOR is monitoring a large number of target devices, it is unlikely that they are all swapped, so it may not terminate automatically. FDRPAS MONITOR tasks with DYNMON=YES only terminates automatically based on the DURATION= operand, if specified. If DURATION= is not specified, then they do not terminate automatically.

You can specify a DURATION=*nn* operand on a MONITOR statement. When the MONITOR task has accumulated "*nn*" minutes of idle time (during which it is not participating in the swap of any volume), it terminates automatically.

FDRPAS also supports the console STOP command (abbreviated P), specifying the job name or started task name of an FDRPAS SWAP or MONITOR task. For example:

```
P MONITOR1
```

If you STOP (P) a MONITOR task, it terminates within a few seconds if it is idle. If it is participating in one or more active swaps, those swaps are allowed to complete, but the new swap requests are not accepted.

If you STOP (P) a SWAP task, all active swaps are allowed to complete, but any requested volumes that have not yet started do not start. Messages are issued to identify the volumes that were bypassed because of the STOP (P).

If you must terminate active swaps for some reason, take these steps in this order, until the swaps are terminated.

1. Issue a STOP (P) command to the SWAP task, which allows currently active volumes to complete. If you cannot wait for active swaps to finish, use the ISPF panels to ABORT the active swaps or issue a CANCEL (C) command. When all swaps have terminated, you can issue STOP (P) commands to the MONITOR tasks if they have not already terminated.
2. If STOP (P) does not work, issue a console CANCEL (C) command to the SWAP task. When all swaps have terminated, you can issue STOP (P) commands to the MONITOR tasks if they have not already terminated.
3. If the SWAP task does not terminate, then issue a CANCEL (C) command for each MONITOR task. Because of cancel protection (see "CANCELPROT=" in Section 310.2), you actually need to issue two CANCEL (C) commands for each task. However, the SWAP task, if still active, does not know that the MONITOR tasks have terminated immediately; they continue copying data until the end of the current copy pass, at that point each swap fails because of the missing MONITOR tasks.

A CANCEL (C) command causes the FDRPAS SWAP or MONITOR task to enter a cleanup routine for each active swap. It may take a minute or so to cleanup all of the active subtasks, so FDRPAS may not terminate immediately. Since a second CANCEL (C) causes the cleanup to be bypassed, which may leave active volumes in an unknown state, FDRPAS rejects any more CANCEL (C) commands while it is in this cleanup until two minutes have passed. After two minutes, another CANCEL (C) is accepted, allowing you to terminate FDRPAS even when it is hung in the cleanup routine. If the CANCEL (C) commands do not work, you can use the console FORCE command to terminate the FDRPAS address space.

WARNING: ***If possible, contact INNOVATION DATA PROCESSING before using a second CANCEL (C) or FORCE to terminate FDRPAS. In any case, contact INNOVATION DATA PROCESSING after such action, since we may need to guide you through a cleanup process.***

FDRPAS CONSOLE STATUS DISPLAYS

You can display the status of the active volumes in an FDRPAS SWAP task on the console by issuing the console `MODIFY (F)` command like this:

```
F jobname,STATUS     (or just STA)
```

FDRPAS responds with messages on the console and in the job log of the FDRPAS job or started task with the status of any volumes currently being swapped, similar to the information displayed by the FDRPAS ISPF interface. For example,

F *JOBNAME*,STATUS

```
FDRW08  VOLSER UNIT TARG  %  PASS TOCOPY COPIED UPDATE STATUS
FDRW08  ------ ---- ---- --- ---- ------ ------ ------ --------------
FDRW08  SH20CC 20CC 20CB  10    1  15017   1545      0 ACTIVE SWAP
FDRW08  SH20C6 20C6 20C5  19    1  15078   2865      0 ACTIVE SWAP
```

RESET SERVICE CLASS

You can reset the service class of an FDRPAS job or started task during execution by issuing a console `MODIFY (F)` command”

```
F jobname,SRVCLASS=classname
```

300.4 FDRPAS ON A SINGLE SYSTEM

SINGLE SYSTEM OPERATION

When only a single z/OS system image (CPU or LPAR) can access the DASD volume to be moved, FDRPAS operation is simple:

- You start an FDRPAS SWAP task for each source volume to be moved, specifying an offline DASD volume as the target device. The SWAP task also acts as a MONITOR task.
- FDRPAS copies tracks from the source volume to the target device.
- During the copy, FDRPAS monitors all I/O operations to the source volume and notes all tracks that have been updated. Updated tracks are copied (or re-copied, if they were previously copied) to the new device.
- When the copy is complete or the number of tracks remaining to be copied is below a threshold, FDRPAS quiesces all I/O to the source volume. The remaining tracks, if any, are copied while all other I/O is quiescent. At this point, the target device is an exact copy of the source volume.
- FDRPAS swaps all system pointers so that all future I/O to the volume is directed to the target device. The original device is placed offline and the volume label on that device modified so that it cannot be accidentally placed online.
- I/O to the volume is re-enabled and the FDRPAS SWAP task terminates.

WARNING: ***You must be sure that the volume being swapped is not online to any other system or LPAR. If it is, you must treat this as a multi-system swap as described in the following section.***

300.5 FDRPAS ON MULTIPLE SYSTEMS

MULTI-SYSTEM OPERATION

When multiple z/OS system images can access the DASD volume to be moved, there are some additional steps, since the swap must be coordinated on all system images. All system images must be monitored for updates to the volume during the swap, and the final swap to the new device must be conducted simultaneously on all images. The sequence is:

- You must ensure that an FDRPAS MONITOR task is running on all systems that have access to the target device (even if it does not have the source volume online) or add the PASJOB DD statement with the appropriate control statements to start the MONITOR tasks with the SWAP job. Each MONITOR task can be directed to monitor only a single target device, a range of potential target devices, or have the target devices added dynamically. A DASD device can connect to up to 128 systems, so FDRPAS supports up to 127 MONITOR tasks for a given SWAP.
- You start the FDRPAS SWAP task on any system, specifying the volume to be swapped and the output (target) device. For best performance, the SWAP task should run on the system with the highest level of update activity on the volume to be swapped.
- After validating the swap request, the FDRPAS SWAP task indicates that the swap is pending.
- On the other system images, the FDRPAS MONITOR tasks recognizes that the swap is pending and indicate that they are ready to participate in the swap. If the MONITOR task is monitoring only a single target device, that task handles the entire swap process. If the MONITOR task is monitoring multiple target devices, the MONITOR task starts a separate FDRPAS task for each volume when the swap begins.
- When the required number of MONITOR tasks have acknowledged their participation, the SWAP task signals that the swap has begun. The SWAP task installs the I/O intercept on its image to monitor updates.
- The MONITOR tasks recognizes that the swap has begun and install the I/O intercept on their images to monitor updates.
- When all MONITOR tasks have indicated that the intercepts are installed, the SWAP task begins copying tracks from the original device to the target device.
- The FDRPAS intercepts on each system monitor all I/O operations to the original device and note all of the tracks that have been updated. Updated tracks are copied (or re-copied, if they were previously copied) to the new device.
- When the copy is complete or the number of tracks remaining to be copied is below a threshold, FDRPAS signals all MONITOR tasks to quiesce all I/O to the original device. The remaining tracks, if any, are copied while all other I/O is quiescent. At this point, the target device is an exact copy of the source volume.
- The SWAP task now signals all MONITOR tasks to swap all system pointers on all system images so that all future I/O to the volume is directed to the new device. The original device is placed offline and the volume label on that device modified so that it cannot be accidentally placed online.
- I/O to the new device is re-enabled, all I/O intercepts are removed, and the SWAP task terminates.

SYSTEM DETERMINATION

In a multi-system environment, one or more FDRPAS MONITOR tasks must be executed on every system image that has the source volume online; one of those MONITOR tasks must monitor the target device if it is in the I/O configuration of that system. If some systems are excluded, those systems are not aware that FDRPAS has moved the volume to a new device, and FDRPAS is not aware of updates to the volume that occur on the excluded systems during the swap. *This could have serious consequences, including data corruption and data loss.*

If you have systems in your complex that have the source volume online but *do not have access to the target device***, you must not attempt to swap the volume to that device.**

FDRPAS attempts to determine how many systems have access to the source volume, in order to protect you against potentially disastrous errors in setting up the FDRPAS swaps. Depending on the DASD hardware involved, FDRPAS may be able to identify the number of systems accessing the source volume and the CPU serial number of each system. However, if the number of systems cannot be determined, or if you need to exclude certain systems from participating in the swap of a given volume, you need to provide input to FDRPAS. Here are the steps that FDRPAS takes:

- On certain DASD subsystems, including the IBM 3990-6, IBM 2105 / IBM 2107 / DS6000 / DS8000, and those that emulate them, FDRPAS is able to determine how many system images have access to the source volume (although it cannot tell if the volume is online or offline). FDRPAS also knows the CPU serial number of each system.
- On most EMC Symmetrix subsystems, FDRPAS is able to tell what systems actually have the source volume online.
- On subsystems where FDRPAS is unable to determine the number of systems accessing the source volume, including the IBM RVA, early Oracle StorageTek SVA and those that emulate a 3990-3, you should contact INNOVATION DATA PROCESSING for documentation on how to migrate this type of DASD with FDRPAS.
- Once the SWAP task signals that the swap is beginning, the MONITOR tasks on each system registers their participation. The SWAP task verifies that the proper number of systems are participating. If the CPU serial numbers of the systems are known, it verifies the serial number of each MONITOR task against the list of expected serials.
- If the expected number of systems (or CPU serials) do not participate, then FDRPAS issues message FDRW68 indicating this condition. If you reply “YES”, the swap continues despite the discrepancy. If “NO” is replied, the swap is terminated (if you specify `NONRESPONDING=FAIL`, then a reply of “NO” is assumed and no FDRW68 message is issued). You may also reply “RETRY”, which causes FDRPAS to wait some additional time to see if the expected number of systems finally participate. The FDRW68 message can be issued as a WTOR to the system operator or you can display and reply to the message from the FDRPAS ISPF panels. **Do not reply “YES” without carefully verifying that all necessary systems are participating; failure to do so may result in data loss or corruption. You should try replying “RETRY” at least once, in case some MONITOR tasks were delayed. Contact INNOVATION DATA PROCESSING before you respond “YES” to an FDRW68 message.**

In the most common configuration, where the source volume and the target device are in the I/O configuration of every system in your complex, you simply need to start a MONITOR task for the output device on every system, and the rest is automatic. If FDRPAS identifies systems that did not register, then the MONITOR task is not executing on those systems; just fix that error and try again.

The process is more complex when the source volume and/or the target device are not in the I/O configuration of some of your systems, or the source volume is offline on some systems, but even then, FDRPAS attempts to automate the process:

- If the source volume is not in the configuration or is offline on some systems, but the target device is in the configuration, you should execute a MONITOR task on

those systems. The MONITOR task sees the swap request, determine that it does not need to participate in the swap because the source volume is not in use, and communicate that to the SWAP task. The SWAP task counts this as a responding system but excludes it from swap processing.

- If the target device is not in the configuration of some systems, but those systems are connected to the system executing the SWAP task via GRS (a GRS complex), then you should execute a MONITOR task with DYNMON=YES on those systems. FDRPAS uses a series of cross-CPU enqueues (major names FDRPAS, and FDRPASQ) to communicate that those systems do not need to participate.

WARNING: ***If some systems have the source volume online but do not have access to the target device, do not attempt to swap that volume unless you vary it offline on those systems first. It is not accessible on those systems after the swap.***

Only in the situation where some systems have the source volume offline but do not have access to the target device and are not connected to the swapping system by GRS or MIM, do you need to take special actions to allow FDRPAS to continue. This also applies if some systems in your configuration are running non-z/OS systems such as z/VM, Linux. or Linux for System z. If the DASD subsystem is a 3990-6, 2105, 2107, or another that allows FDRPAS to determine the CPU serial numbers of the systems accessing the source volume:

- You can use the `EXCLUDE CPUID=` statement of FDRPAS (see Section 310.4 "SWAP Task EXCLUDE Statement") to specify the serial numbers of the systems that do not have the source volume online. **EXCLUDE is the recommended way of handling such systems.**
- If you do the above, but you still get the FDRW68 message indicating that there are non-responding systems, you should reply "RETRY" at least once to be sure that a slow system was not prevented from replying. If the FDRW68 is reissued, then you should reply "NO" to terminate the swap, investigate the cause, and update the FDRPAS input statements or start the proper FDRPAS MONITOR tasks to correct the error. **Do not reply "YES" to the FDRW68 unless you are absolutely certain that all necessary systems are participating.**

Therefore, in many installations, all devices in all DASD subsystems are defined to all systems in the complex, so executing FDRPAS is simply a matter of making sure that proper FDRPAS MONITOR tasks are running on every system.

In some installations, such as service bureaus and outsourcing sites, certain devices in DASD subsystems may be deliberately omitted from the I/O configuration on some systems, to prevent inadvertent access. In these installations, more care must be taken to be sure that the requirements for FDRPAS are met.

300.6 FDRPAS HISTORY RECORDS

HISTORY RECORDS (OPTIONAL)

So that you can display a history of the swaps that FDRPAS has performed, FDRPAS creates a history record for every successful swap. These history records are simple catalog entries in an ICF catalog. No special database is required. These names exist only in the catalog; no real data sets by these names are created.

The high-level index of the FDRPAS history records is the value specified for PASINDEX in the FDR option table in the FDRPAS load library. By default, the value of PASINDEX is "FDRPAS".

To record history records, you need to define an ICF user catalog (or choose an existing catalog to use) and associate an alias to that catalog in your master catalog; the alias name must match the value of PASINDEX. This catalog can be shared among the systems swapping the volumes, or you can define a unique catalog on every system (or a mixture). FDRPAS SWAP and MONITOR tasks must have authority to create data sets starting with the PASINDEX into the aliased user catalog. If they do not, the history records are not created but the swaps run successfully.

Do not assign PASINDEX to a user catalog with a lot of activity. Because every SWAP and MONITOR task attempts to catalog into that catalog when a swap ends, there are catalog updates from multiple systems in a very short period of time. Depending on the type of catalog sharing and caching in use, this can cause other catalog requests to be delayed and may cause FDRPAS to be delayed. If desired, you can DEFINE a new user catalog for the PASINDEX.

If you choose not to record FDRPAS history records on one or more systems, then do not define an alias matching PASINDEX in the master catalog of those systems. You receive a message indicating that the alias does not exist, but the swap ends normally.

The format of the cataloged data set name is:

```
pasindex.Svolser.Dyyyyddd.Thhmmss.sysname
```

This documents the volume serial that was swapped, the date and time of the swap, and the name of the system it was swapped on. If multiple systems are involved, a separate history record is created for each. For example,

```
FDRPAS.SPROD01.D2013003.T091242.PRODSYS
```

This indicates that volume PROD01 was swapped on 2013/003 (January 3, 2013) at 09:12:42 on system PRODSYS.

Other fields in the catalog entry record the original device address of the volume, and the device address it was swapped to. Although you can display these catalog records with IDCAMS LISTCAT and other utilities, the internal fields with the device addresses are not displayed by those utilities.

Note that system names can be any eight alpha, numeric, or national characters; they do not have to start with an alphabetic character. However, if the first character is numeric, it creates a nonstandard data set name, which would be rejected when FDRPAS attempted to catalog it. In such cases, FDRPAS sets the last index level of the cataloged dsname to "#" (number sign) followed by the first seven characters of the system name. If the system name is seven or less characters, the full system name is used after the "#" (number sign). (Customers outside the US: the # character can be displayed with whatever character generates hex value x'7B' in the local language).

Without further action, history records are retained indefinitely, but you can discard them simply by uncataloging them. You can automate deletion of old history records by using the `HISTORY TYPE=SWAP` command of FDRPAS (see Section 310.10 "MONITOR WAITSWAPTERM MOUNT Statement").

You can display the history records with HISTORY command of the FDRPAS ISPF panels (see Section 310.31 "FDRPAS ISPF Interface"). Here is a sample:

FDRPAS HISTORY – PANEL A.E – HISTORY COMMAND

```
--------------------------- FDRPAS Plug & Swap History  --- Row 1 to 3 of 3
COMMAND ===>                                              SCROLL ===> PAGE

Command  Volume  Unit  Swapped    System     Date        Time
         Serial  Addr  to Unit
-------  ------  ----  -------   --------  ----------  --------
         PROD01  17CC  3BCC      CPUA      12/22/2012  16:42:34
         PROD01  17CC  3BCC      CPUB      12/22/2012  16:42:35
         PROD01  17CC  3BCC      CPUC      12/22/2012  16:42:32
```

305 FDRPAS GENSWAP JCL, STATEMENTS, AND EXAMPLES

305.1 GENSWAP OVERVIEW

FDRPAS GENSWAP provides the ability to generate the JCL and control statements for both the main task and the MONITOR task(s) to ensure they are consistent and in agreement. GENSWAP is designed for everything from simple SWAP operations to swapping large numbers of volumes concurrently.

The improvements in the design and better integration with system services allows FDRPAS to be more robust and provide the ability to process a large number of volumes concurrently, providing substantial savings in time to swap volumes.

JUST-IN-TIME MONITORS

SIMSWAPMON/SWAP/SWAPDUMP jobs that are generated by GENSWAP use "just-in-time" monitors. The traditional way to run FDRPAS was to start the monitors some time before the main job; typically a monitor was set up to monitor a large range of potential target volumes, and a given monitor would respond to a number of main jobs. When using GENSWAP, each main job submits a monitor job to all of the necessary LPARs, and passes to it a list of SWAPUNITs containing only those target units that are used by this main job. The overhead of the monitor, although small to begin with, is further reduced by not monitoring unnecessary units. The monitor automatically terminates shortly after the SWAP job completes.

Dynamic monitoring was an earlier approach to just-in-time monitors. With dynamic monitoring, the main job submits a monitor job to all of the necessary LPARs but does not pass any list of SWAPUNITs to the monitors. The main job uses enqueues to communicate the SWAPUNITs to the monitors. Dynamic monitoring requires all of the LPARs to be in the same GRS or MIM complex; the GENSWAP approach works whether or not the LPARs are in a GRS or MIM complex. The enqueue communication for dynamic monitoring has a certain amount of overhead. With the GENSWAP approach, the monitor starts off being able to respond to dynamic monitoring, but discontinues the enqueue communication for target units as soon as SWAP tasks are active on all of the SWAPUNITs that were passed to it.

Under the GENSWAP approach, If there is an LPAR that is connected to the source volume but cannot access the target device the monitor responds by using the dynamic-monitoring type of enqueue. In this case, the SWAP can only proceed if the source volume is offline on that monitor LPAR. If the source volume is online on an LPAR and the monitor on that LPAR cannot access the target device, the SWAP fails.

If there is an LPAR that is connected to the source volume but cannot access the target device, and that LPAR is not in a GRS or MIM complex with the main LPAR, and you can guarantee that the source volume is offline on that LPAR, then, in the main job, you can supply an `EXCLUDE CPUID=` statement for that LPAR (see Section 310.4 "SWAP Task EXCLUDE Statement"). If the source volume is online on that LPAR, then the SWAP cannot be done successfully.

QUERY HOST ACCESS SUPPORT

Query Host Access (QHA) is supported and utilized for the storage subsystems that support this function. This function gives FDRPAS a list of all the LPARs that are online to a volume allowing FDRPAS to ensure that monitor tasks are running on all the necessary LPARs. With this support, any SWAP task `EXCLUDE CPUID=` statements (see Section 310.4 "SWAP Task EXCLUDE Statement") are ignored on storage subsystems that support QHA and are online.

- **IBM Storage Subsystems** – Query Host Access is available for the DS8870 in code Release 7.1, GA 6/07/2013, and we expect that QHA will soon become available for the DS8700 and DS8800. We urge all FDRPAS customers to install these microcode levels when available, to support enhanced validation by FDRPAS.
- **EMC Storage Subsystems** – FDRPAS already supports this feature in EMC storage subsystems through the Logical Path Query command.

305.2 FDRPAS GENSWAP JOB CONTROL REQUIREMENTS

The following JCL statements are required to execute FDRPAS GENSWAP tasks as batch jobs or started tasks.

EXEC STATEMENT

Specifies the program name (`PGM=FDRPAS`), region requirement (`REGION=0M` is recommended), and optional `PARM=` field that may contain the first FDRPAS control statement. For example,

```
//GENSWAP  EXEC PGM=FDRPAS,REGION=0M,
//          PARM='GENSWAP TYPE=FULL,…'
```

STEPLIB OR JOBLIB DD STATEMENT

Specifies the load library where FDRPAS resides. The library must be authorized. FDRPAS must be executed with a JOBLIB or STEPLIB, it should never be put into the system linklist.

SYSPRINT DD STATEMENT

Specifies the output message data set; it is required. It is usually a SYSOUT data set but if it is assigned to a data set on tape or DASD, this DD statement must specify `DISP=MOD`. DCB characteristics are `RECFM=FBA` and `LRECL=121`; the block size defaults to 1210 on DASD or tape.

FDREMAIL DD STATEMENT

Specifies input control statements for the FDR e-mail facility. If present, e-mail messages can be sent for unsuccessful or successful FDRPAS operations. See Section 320.4 "FDRPAS E-mail Notification Facility" for requirements and details.

SYSUDUMP DD STATEMENT

Specifies the ABEND dump data set. Usually specifies a SYSOUT data set. Although the SYSUDUMP DD statement is not required, we strongly urge you to always include this DD statement, so that we can help you diagnose error conditions. If you have a debugging aid product on your system that would prevent the desired dump, please add the appropriate one of these statements to the JCL so that a fully-formatted dump is produced.

```
//ABNLIGNR DD DUMMY  Turn off Abend-Aid
//CAOESTOP DD DUMMY  Turn off CA OPT II & CA SYMDUMP
//DMBENAN  DD DUMMY  Turn off DumpMaster
//ESPYIBM  DD DUMMY  Turn off Eye-Spy
//IDIOFF   DD DUMMY  Turn off IBM Fault Analyzer
```

INTRDR DD STATEMENT

Specifies where the generated control statements are written. If not specified, FDRPAS dynamically allocates INTRDR to the internal reader for job submission.

Specify `SYSOUT=*` if you wish to review the generated control statements before they are submitted.

Specify a sequential data set or a member of a partitioned data set if you wish to save the generated output.

SYSIN DD STATEMENT

Specifies the control statement data set. Usually an input stream or `DD *` data set.

PASJOB DD STATEMENT

Specifies the input control statement for the FDRPAS job(s) GENSWAP submission facility. Usually specifies `DD DATA,DLM=xx` options. The `DLM=` specifies two (2) unique characters that delineate the end of the job stream for each PASJOB DD. "`SS`" is recommended for the main SWAP jobs and "`MM`" for the Monitor jobs.

The PASJOB DD is followed by the control statements for the SIMSWAP, SIMSWAPMON, SWAP, or SWAPDUMP job that is to be generated. This is the outer layer of control statements. Within the outer layer, there may be another PASJOB DD (the inner layer) that specifies the MONITOR control statements that are generated when the actual SIMSWAPMON, SWAP, or SWAPDUMP job is executed.

The outer layer PASJOB DD is followed by the JOB statement and PASPROC procedure and SYSIN control statements as documented under the appropriate FDRPAS command (ex; `SWAP TYPE=FULL` is in Section 310.2 "SWAP, SWAPDUMP, SIMSWAP, SIMSWAPMON Statements"). If no special routing or class is specified, the job is executed on the same LPAR as the GENSWAP. Optionally, a list of classes or routing options can be specified that is used in rotation if more than 1 job is generated. This allows the jobs to be executed on multiple LPARs.

The inner layer PASJOB DD is followed by the JOB statement and PASPROC procedure and SYSIN control statements as documented under the FDRPAS Monitor command (Section 310.5 "MONITOR SWAP Statement"). Each JOB statement is preceded by a `//*CPUID` statement that specifies the CPUID the monitor job is to execute. One MONITOR job must be specified for each LPAR that is needed for the SWAP.

The job name and some other control operands can have ampersands (&) specified within them. FDRPAS substitutes a unique number in each job stream that is generated. Depending on the number of jobs needed the user can specify 1 to 3 ampersands in a row. For example, `//PASSW&& JOB …` gets a unique 2-digit number for each generated job.

305.3 GENSWAP STATEMENT

GENSWAP STATEMENT SYNTAX

The syntax of the GENSWAP statement:

```
GENSWAP   TYPE=FULL
          ,ALREADYINSWAP=BYPASS          ,MAXGEN=64|nnnnn
          ,MAXACTIVESWAPS=NO|nnn         ,SUBMITDELAY=15|nn
          ,MAXCARDS=9999|nnnnn           ,SWAPID=nnn
          ,MAXDD=1000|nnnnn              ,VOLSORT=NO|YES
```

GENSWAP STATEMENT

This statement is used to generate the job stream for a subsequent FDRPAS job. The model JCL and control cards for the job stream are in the PASJOB DD (input), and the generated job stream is written to the INTRDR DD. If an INTRDR DD statement is not supplied, then by default, GENSWAP dynamically allocates INTRDR to a JES internal reader and submits the generated job stream for immediate execution. Alternatively, INTRDR can be allocated to SYSOUT or to a data set, for checking and possible later execution. Within the generated job stream for an FDRPAS SIMSWAPMON, SWAP, or SWAPDUMP job, there can be embedded job streams for FDRPAS monitors to be submitted to other LPARs.

If you set up a SWAP, SWAPDUMP, SIMSWAP, or SIMSWAPMON manually, you have to code a separate MOUNT statement for every volume to be swapped, with a corresponding SWAPUNIT. With GENSWAP, however, you can select all of the volumes in an SSID or in an entire control unit with one MOUNT statement. This works well when there is a correspondence between the device addresses of the source and target units. For example, if the source volumes are in an SSID with the address range 1000 to 10FF, and the target devices are in the address range A000 to A0FF, then the target units can be specified as `SWAPUNIT=A0**` or `SWAPUNIT=A***`. With GENSWAP, you can also EXCLUDE volumes that are within the scope of a following SELECT statement.

GENSWAP generates one or more jobs; the number of volumes included in each job is specified by the `MAXGEN=` operand. If more than one job is to be generated, then to make the job names unique, the model JOB statement in PASJOB must include one, two, or three ampersands (`&`) to be used as substitution characters.

GENSWAP can also generate jobs to execute across multiple LPARs.

GENSWAP STATEMENT OPERANDS

TYPE=FULL

Must be specified on the GENSWAP statement.

ALREADYINSWAP=BYPASS

Do not include in the generated MOUNT command a volume for which a SWAP or SWAPDUMP is currently in progress. This operand allows a rerun of the GENSWAP job using the same GENSWAP MOUNT statement(s) and only generates the SWAP and SWAPDUMP MOUNT statement(s) for the volume(s) in the job that failed, were aborted, or were excluded from an earlier run. Refer to "Dealing with Possible Problems" in Section 305.9 for additional information on using this operand.

NOTE: Only use this operand with `CONFIRMSWAP=YES` or `CONFIRMSPLIT=YES`.

MAXACTIVESWAPS=

NO – GENSWAP does not control the MAXACTIVESWAPS option for the generated jobs. The `MAXACTIVESWAPS=` operand on the SWAP or SWAPDUMP statement in the generated jobs is honored.

***nnn* –** The number of FDRPAS tasks that can be in the initial copy phase within **one** LPAR is limited to "*nnn*" tasks. Values for "*nnn*" may be from 1 to 128. GENSWAP adds the keyword `MASID=` ("max active swap ID"; no relation to the MASID operand of the ENQ macro) to the generated MOUNT statements in the SWAP or SWAPDUMP jobs. The value of `MASID=` ranges from 1 to "*nnn*". Only one FDRPAS task with a given `MASID` is allowed to be in the initial copy phase on one system at any given time.

`MAXACTIVESWAPS=`*nnn* is especially useful when `CONFIRMSWAP=YES` or `CONFIRMSPLIT=YES` is specified (in the SWAP main task), since this usually implies many volumes are in "confirm" state before the operation is finally confirmed and terminated. This allows you to submit the FDRPAS jobs for all those volumes at once knowing that no more than "*nnn*" are ever in "pass 1" **within one LPAR**.

Default: NO.

MAXCARDS=

***nnnnn* –** Specifies the maximum number of MOUNT statements that can be generated in this GENSWAP step, from 1 to 32767.

Default: 9999.

NOTE: It should seldom be necessary to specify this operand.

MAXGEN=

***nnnnn* –** Specifies the number of volumes (1 to 32767) to be included in each generated job. Each time this number of MOUNT statements is generated:

- GENSWAP copies any remaining records in the PASJOB input
- submits the job
- returns to the beginning of its PASJOB input
- increments the job ID number
- starts generating a new job

Default: 64.

SUBMITDELAY=

***nnnn* –** Specifies the time interval in seconds, from 0 to 3600, that GENSWAP waits between submitting jobs. The purpose of this operand is to avoid flooding the system with jobs and to allow the SWAP tasks to start in a more orderly manner, when GENSWAP is submitting SIMSWAPMON, SWAP, and SWAPDUMP jobs to the internal reader.

`SUBMITDELAY=` is ignored when:

- The INTRDR DD is allocated to SYSOUT or a data set; in this case, GENSWAP does not wait between jobs.
- When the job being submitted is SIMSWAP; in this case GENSWAP does not wait between jobs (because SIMSWAP jobs run quickly and do not have to coordinate with monitors).

Default: 15.

SWAPID=

nnn **–** Specifies a starting value (1 to 255) for a sequential number to be associated with each generated job. When the number of volumes specified by MAXGEN has been included in a generated job, GENSWAP:

- copies any remaining records in the PASJOB input
- submits the job
- returns to the beginning of its PASJOB input
- increments the SWAPID number
- starts generating a new job

NOTE: When SWAPID is incremented, the value 64 is skipped. Also, if SWAPID=64 is specified, it is treated as 65. (64 is X'40' = blank, which has special meaning internally.)

The `SWAPID=` value is used for several purposes:

- It is substituted into the job names of the generated jobs in order to give them unique names. When more than one job is to be generated, the job name on the JOB statement in PASJOB should include one, two, or three ampersands (`&`). GENSWAP replaces the ampersand(s) with the current `SWAPID=` value. GENSWAP fails if there are not enough ampersands (two if `SWAPID=` is higher than 9, and three if `SWAPID=` is higher than 99)
- It is used internally in FDRPAS enqueues.

Default: 1.

VOLSORT=

NO – The GENSWAP process does not sort the volumes being processed by size.

YES – The GENSWAP process sorts all the volumes by size (largest first) and selects one volume of a given size from each separate SSID in rotation, in order to avoid contention.

Default: NO.

305.4 GENSWAP TASK EXCLUDE STATEMENT

GENSWAP EXCLUDE STATEMENT SYNTAX

FDRPAS SWAP task EXCLUDE statement syntax:

EXCLUDE X	VOL=*volspc*

GENSWAP EXCLUDE STATEMENT

The GENSWAP task EXCLUDE statement is optional. If specified, the EXCLUDE statement(s) precede the GENSWAP MOUNT statement(s) and specify volumes to be excluded from the generated output. One or more EXCLUDE statements can be specified.

GENSWAP EXCLUDE STATEMENT OPERANDS

VOL=

volspc – Specifies the volume serial number(s) to be excluded from GENSWAP processing. Volumes can be coded as:

```
EXCLUDE VOL=AB1808            exclude a single volser
EXCLUDE VOL=AB185*            exclude all volumes
                              matching a specific mask
EXCLUDE VOL=(AB1824,AB1825)   exclude a specific
                              list of volumes
```

A maximum of 255 volsers can be specified on a single EXCLUDE statement.

305.5 GENSWAP TASK MOUNT STATEMENT

GENSWAP MOUNT STATEMENT SYNTAX

FDRPAS GENSWAP task MOUNT statement syntax:

MOUNT	VOL=*volspc* ,SWAPUNIT=*uuuu*
	,CUID=*sssss*| SSID=*xxxx*

GENSWAP MOUNT STATEMENT

The GENSWAP task MOUNT statement follows the GENSWAP statement and any GENSWAP EXCLUDE statements specifying online volumes to be swapped. One or more MOUNT statements may specified. At least one MOUNT statement is required.

GENSWAP MOUNT STATEMENT REQUIRED OPERANDS

VOL=

volser **–** Specifies the volume serial number(s) of online volume(s) to be selected for GENSWAP processing. Examples of VOL= statements are:

```
MOUNT VOL=AB1808,SWAPUNIT=A808
MOUNT VOL=AB185*,SWAPUNIT=A85*
MOUNT VOL=(AB1824,AB1825),SWAPUNIT=A82*
MOUNT VOL=*,SWAPUNIT=A***,CUID=12345
```

SWAPUNIT=
SU=

uuuu **–** Specifies the z/OS device address of the target device, the new DASD unit that this volume is swapped to. You must specify this as a 4-digit address, with a leading zero if required. This device should be offline on all systems that are involved in the SWAP process and should not contain any needed data (CHECKTARGET=YES can be used to ensure that the target device is empty).

It is possible to specify an asterisk (*) for any digit of the SWAPUNIT= address. FDRPAS substitutes the corresponding digit of the z/OS address of the source volume (the volume specified by VOL=). For example, if the z/OS address of volume PROD01 is 03A4:

```
MOUNT VOL=PROD01,SWAPUNIT=17C*   swaps to device 17C4
MOUNT VOL=PROD01,SWAPUNIT=17*2   swaps to device 17A2
MOUNT VOL=PROD01,SWAPUNIT=17**   swaps to device 17A4
```

This feature allows you to easily swap a set of volumes on adjacent addresses (a "string" of DASD volumes) to a set of adjacent addresses in the new DASD subsystem.

GENSWAP MOUNT STATEMENT OPTIONAL OPERANDS

CUID=

sssss **–** Limits the source volume(s) selected to those in the specified Control Unit (CU).

Specifying CUID= with VOL=* allows the SWAP of a complete Control Unit as long as the target Control Unit is defined like the source Control Unit with regard to UCB addresses and volume sizes (equal or larger). For example, to move all the volumes in CUID=12345 to a new Control Unit defined by UCB addresses 9***, specify:

```
MOUNT VOL=*,SWAPUNIT=9***,CUID=12345
```

NOTE: CUID= and SSID= are mutually exclusive. CUID=, if specified, must be the last operand on the statement. CUID= must specify 5 characters.

SSID=

xxxx **–** Limits the source volume(s) selected to those in the Subsystem ID specified. SSID must be 4 hexadecimal digits unless SSID=* is specified. SSID=* selects all volumes in any SSID that matches VOL=.

NOTE: CUID= and SSID= are mutually exclusive. SSID=, if specified, must be the last operand on the statement.

305.6 GENSWAP MODEL DEFINITION

The PASJOB DD statement contains the model definition for the generated SWAP jobs and consists of JCL and control statements.

NOTE: The following is a definition of the statements necessary to generate the JCL and control statements for SIMSWAP, SIMSWAPMON, SWAP, and SWAPDUMP as well as the associated MONITOR tasks. Following this definition of statements is a sample GENSWAP job specification with numbered lines showing PASJOBs under a GENSWAP.

We highly recommend using the PASPROC procedure to generate the individual JCL as it simplifies the tasks. See statements (13), (22), and (30).

For example: `//SWAP1  EXEC PASPROC,LIB=fdrpas.loadlib`

SWAPJOB STATEMENT

The `SWAPJOB` statement is used to indicate the start of the main task model (outer layer). This statement must be the first statement and coded as `//*SWAPJOB`. See statement (8).

JOB CARD FOR MAIN JOB STATEMENT

Specifies the JOB statement necessary to execute the job on the correct LPAR. The job name may contain ampersands (`&`) to specify the position in the job name where the job number is incremented to produce unique job names. The starting value for the ampersands is set by the SWAPID= operand on the GENSWAP statement. 1 to 3 ampersands my be specified as needed to generate the correct number of unique jobs. See statement (9) and review "Establish a JOB Naming Convention" in Section 305.6.

Example: `//PASSW&&  JOB (ACCT),'USER NAME',MSGCLASS=X,`

JES3 NOTE: If the job is to be executed on another LPAR, JES3 requires two JOB statements. The second JOB statement should specify NJB instead of JOB.

FDRPAS JOB ROUTING STATEMENTS

Specify the necessary routing information via the `/*ROUTE`, `//*MAIN`, or `/*JOBPARM` statements or the `CLASS=` operand on the JOB card, depending on the procedures used to route jobs at your site. If multiple systems are to receive the main SWAP job, code an ampersand instead of the system value followed by the list of systems where the main SWAP job is to be executed. This coding directs the generated jobs to be routed to all the systems in a circular pattern until all the necessary jobs are generated. See statement (11) for an example using the `/*ROUTE XEQ` routing statement. When a `/*ROUTE PRINT &` statement (12) is coded, it inherits the value from the preceding `/*ROUTE XEQ &` statement.

Examples of routing statements are:

```
/*ROUTE XEQ JESCPUA
/*ROUTE XEQ & JESCPUC,JESCPUB,JESCPUA     <- JES2 routing
//*MAIN SYSTEM=& JESCPUC,JESCPUB,JESCPUA  <- JES3 routing
/*JOBPARM SYSAFF=& CPUC,CPUB,CPUA    <- JES2 multi-access
```

If multiple classes need to specified, code the `CLASS=&` operand on a line by itself within the JOB statement. For example, `// CLASS=& M,N,O` The classes are rotated independent of the routing statement. See statement (10).

NOTE: Only one `/*ROUTE`, `//*MAIN`, or `/*JOBPARM` statement can be specified per JOB. The maximum number of JES target LPARs are the number that fits on a single line up to column 71.

FDRPAS MAIN STATEMENT

This statement specifies the operation to be performed (SIMSWAP, SIMSWAPMON, SWAP, or SWAPDUMP) and its options. See statement (15). All the different operations (SIMSWAP, SIMSWAPMON, SWAP, and SWAPDUMP) for the same set of source volumes can be generated by just changing this one statement and rerunning the GENSWAP job. This is one of the major benefits of GENSWAP.

Example: `SWAP      TYPE=FULL,MAXTASKS=32,ALLOWPAV=YES`

FDRPAS MAIN MOUNT STATEMENT

This is the model used to generate the MOUNT VOL= statements for all the volumes that GENSWAP selects. This control card must be coded as **MOUNT VOL=&&&&&&,SWAPUNIT=&&&&**. GENSWAP generates as many MOUNT statements as are needed for all the volumes selected.

Each generated MOUNT statement is numbered sequentially even if multiple jobs are created (for example, <------1). The highest number that is generated can be used to verify that the correct number of volumes have been selected by the GENSWAP process. See statement (16).

INNER LAYER PASJOB

This DD statement is used to contain the model definition of the MONITOR jobs associated with the main (SIMSWAP, SIMSWAPMON, SWAP, and SWAPDUMP) job. The JCL within this DD statement is used to generate the corresponding MONITOR jobs See statement (18). The model definition for the MONITOR jobs consist of complete JCL to run monitors on all the necessary LPARs as identified by the SIMSWAP job that was previously run.

Example: `//PAS.PASJOB DD DATA,DLM=MM`

NOTE: Repeat the following Inner Layer statements as necessary to generate a MONITOR task for each LPAR. All possible LPARs including the LPAR where this job runs should be included.

CPUID STATEMENT

Used to mark the start of a MONITOR job model and identify the LPAR where this MONITOR job runs. See statements (19), (27), and (34).

Example: `//*CPUID=0912342818`

SWAPNEXT STATEMENT

Used to mark the start of a job that gets the same volume serial numbers as a previous step. The SWAPNEXT statement is mainly used to generate special jobs such as `MONITOR TYPE=CONFIRM` jobs. Example: `//*SWAPNEXT`

JOB CARD FOR MONITOR JOB STATEMENT

This statement specifies the JOB statement for the MONITOR job. If ampersands (&) are specified in the main JOB card statement (9), ampersands (&) should be specified here to ensure unique job names. GENSWAP generates the same unique number as is generated in the main JOB statement. See statements (20), (28), and (35) and review "Establish a JOB Naming Convention" in Section 305.6.

Example: `//PASSMA&& JOB (ACCT),'USER NAME',CLASS=M,MSGCLASS=X`

MONITOR JOB ROUTING STATEMENT

This route must specify the correct value via the `/*ROUTE`, `//*MAIN`, or `/*JOBPARM` statements or the `CLASS=` operand on the JOB card to get the job to the LPAR specified in the preceding `CPUID=` statement. See statements (21), (29), and (36).

Example: `/*ROUTE XEQ JESCPUA`

MONITOR STATEMENT

This statement specifies the MONITOR statement and the options for the MONITOR task. See statements (24), (31), and (39).

Example: `MONITOR TYPE=SWAP,DURATION=2,ALLOWPAV=YES`

MONITOR MOUNT STATEMENT

This is the model used to generate the MOUNT `SWAPUNIT=` statements for all the volumes. This control card must be coded as **MOUNT SWAPUNIT=&&&&**. The SWAP, SIMSWAP, and SWAPMON jobs generate one or more MOUNT statements with all of the target units needed for the corresponding main JOB. See statements (25), (32), and (40).

MONITOR END STATEMENT

This statement marks the end of the MONITOR statements (Inner Layer). Specify the value `MM` to correspond with the value specified on the Inner Layer PASJOB statement (18). See statement (42).

Example: `MM <- End of inner MONITOR templates`

MAIN PROCESS END STATEMENT

This statement marks the end of the main process statements (Outer Layer). Specify the value `SS` to correspond with the value specified on the Outer Layer PASJOB statement (7). See statement (43).

Example: `SS <- End of outer PASJOB template`

RUN FDRPAS MONITOR AS A STARTED TASK

Monitor jobs can be executed as jobs or started tasks. If you are swapping thousands of volume at the same time (CONFIRMSWAP), the monitors can run as started tasks so all your initiators are not tied up. Substitute this JCL for the MONITOR PASPROC if you want the MONITOR to be a started task instead of a job. This replaces statements (22) through (26), (30) through (33), and (37) through (41) in the following sample procedure.

Program FDREMCS is used to:

1. copy the MONITOR control statements from SYSUT1 to a SYSUT2 data set on DASD on the LPAR where the MONITOR task runs
2. issue a start command for the MONITOR on the LPAR where the MONITOR task runs

FDREMCS then terminates and the output is returned to the originating LPAR or as directed by the `/*ROUTE PRINT` statement.

The data set name on the SYSUT2 DD must include "`&JOBNAME`". GENSWAP substitutes the job name from the JOB statement (statements (20), (28) and (35)) so that each started task inherits the same name as the FDREMCS job, ensuring that each of the started tasks generated from this GENSWAP procedure has a unique name. The SYSOUT from the MONITOR task stays on the LPAR where the MONITOR runs.

```
//SUBMONA  EXEC PGM=FDREMCS
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib  (a)
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSUT1   DD *
  MONITOR  TYPE=SWAP,DURATION=2,ALLOWPAV=YES
  MOUNT    SWAPUNIT=&&&&
/*
//SYSUT2   DD UNIT=temp_unit_name,  (b)
//            DSN=fdrpas.&JOBNAME.input.to.monitor,  (c)
//            DISP=(,CATLG),SPACE=(TRK,(1,1)),
//            DCB=BLKSIZE=3200
//SYSIN    DD *
 COMMAND=S PASPROC,JOBNAME=&JOBNAME,
         LIB=fdrpas.loadlib,  (d)
         IN=fdrpas.&JOBNAME.input.to.monitor,  (e)
         D='(OLD,DELETE),FREE=CLOSE'
/*
```

(a) *fdrpas.loadlib* – change this to the name of the FDRPAS load library.
(b) *temp_unit_name* – change this to a valid unit name.
(c) *fdrpas*.&JOBNAME.*input.to.monitor* – change this to a valid data set name. (&JOBNAME must be coded as a node in the data set name). This data set can be SMS-managed.
(d) *fdrpas.loadlib* – change this to the name of your FDRPAS load library.
(e) *fdrpas*.&JOBNAME.*input.to.monitor* – change this to a valid data set name (must be the same name as specified on the SYSUT2 statement).

ESTABLISH A JOB NAMING CONVENTION

Numerous jobs can be generated by the GENSWAP procedure. Because of this, a JOB naming convention should be setup for the main jobs and monitor jobs to ensure unique names and to avoid contention and confusion. The examples in this section of the manual use the following JOB naming convention. (You can use any naming convention, but should try to incorporate the naming conventions for item #2 identifying the type of FDRPAS job in a set location of the jobname.)

1. Characters 1 to 3 contain “PAS” to identify these as FDRPAS jobs.
2. Characters 4 and 5 are used to identify the type of FDRPAS job.
   - **GS** – for GENSWAP job.
   - **SS** – for SIMSWAP jobs.
   - **SM** – for SIMSWAPMON jobs.
   - **SW** – for SWAP jobs.
   - **SD** – for SWAPDUMP jobs.
   - **M*x*** – for MONITOR jobs.
   - **C*x*** – for CONFIRM job.
3. For MONITOR and CONFIRM jobs, the ‘*x*’ value is a unique character to identify the LPAR where the job runs.
4. The next 1 to 3 characters are ampersands (&) for the generated SIMSWAPMON, SWAP, SWAPDUMP, and MONITOR jobs. The GENSWAP process substitutes the SWAPID value to create sets of jobs where the MONITOR jobs have the same number as the corresponding main job. (The ampersands can be anywhere in the job name that allows a numeric value.)

SAMPLE GENSWAP PROCEDURE

This is a sample procedure with numbered lines showing PASJOBs under a GENSWAP. The numbered lines correspond with the references mentioned in the previous definitions.

```
1  //GENSWAP  EXEC PASPROC,LIB=fdrpas.loadlib
2  //PAS.INTRDR DD SYSOUT=*
3  //PAS.SYSIN  DD *
4    GENSWAP  TYPE=FULL,MAXGEN=200,VOLSORT=YES
5    MOUNT    VOL=*,SWAPUNIT=5***,CUID=43928
6  /*
7  //PAS.PASJOB DD DATA,DLM=SS
8  //*SWAPJOB
9  //PASSW&&   JOB (ACCT),'USER NAME',MSGCLASS=X,
10 // CLASS=& M,N,O
11 /*ROUTE XEQ & JESCPUC,JESCPUB,JESCPUA
12 /*ROUTE PRINT &
13 //SWAP1    EXEC PASPROC,LIB=fdrpas.loadlib
14 //PAS.SYSIN  DD *
15   SWAP     TYPE=FULL,MAXTASKS=32,ALLOWPAV=YES
16   MOUNT    VOL=&&&&&&,SWAPUNIT=&&&&
17 /*
18 //PAS.PASJOB DD DATA,DLM=MM
19 //*CPUID=0912342818
20 //PASSMA&& JOB (ACCT),'USER NAME',CLASS=M,MSGCLASS=X
21 /*ROUTE    XEQ JESCPUA
22 //MONA     EXEC PASPROC,LIB=fdrpas.loadlib
23 //PAS.SYSIN  DD *
24   MONITOR  TYPE=SWAP,DURATION=2,ALLOWPAV=YES
25   MOUNT    SWAPUNIT=&&&&
26 /*
27 //*CPUID=0212342818
28 //PASSMB&& JOB (ACCT),'USER NAME',CLASS=M,MSGCLASS=X
29 /*ROUTE    XEQ JESCPUB
30 //MONB     EXEC PASPROC,LIB=fdrpas.loadlib
31   MONITOR  TYPE=SWAP,DURATION=2,ALLOWPAV=YES
32   MOUNT    SWAPUNIT=&&&&
33 /*
34 //*CPUID=0112342818
35 //PASSMC&& JOB (ACCT),'USER NAME',CLASS=M,MSGCLASS=X
36 /*ROUTE    XEQ JESCPUC
37 //MONC     EXEC PASPROC,LIB=fdrpas.loadlib
38 //PAS.SYSIN  DD *
39   MONITOR  TYPE=SWAP,DURATION=2,ALLOWPAV=YES
40   MOUNT    SWAPUNIT=&&&&
41 /*
42 MM                  <- End of inner MONITOR templates
43 SS                  <- End of outer PASJOB template
```

Note: Use the **Alt+Left Arrow** keys, to return to previous location.

305.7 GENSWAP PROCEDURE AND EXAMPLES

This section gives examples of GENSWAP operations that show how FDRPAS JCL and control statements can be generated for operations ranging from very basic SWAP tasks to complex large concurrent SWAP tasks running multiple SWAP tasks on multiple systems with corresponding MONITOR tasks. All examples in this section are found in the JCL library installed with FDRPAS. The member names are PA30507*x*.

RECOMMENDATION: *Use the PASPROC procedure to ensure that all the tasks use the same JCL and to reduce the number of JCL statements needed.*

GENSWAP PROCEDURE

The recommended sequence of operations for GENSWAP is:

1. SIMSWAP
2. SIMSWAPMON
3. SWAP or SWAPDUMP
4. CONFIRM (if `CONFIRMSWAP=` or `CONFIRMSPLIT=` was specified in the SWAP or SWAPDUMP job(s).)

The MOUNT statements for GENSWAP specify the desired set of source volumes and stay the same throughout the sequence. The SIMSWAP starts off with JCL to run on one system. After the SIMSWAP, add the CPUID statement and the JCL for monitors to run on every system that is connected to the source volumes as identified by the SIMSWAP job. Then, run the SIMSWAPMON to validate the monitor jobs and to ensure that all needed LPARs respond. Then, just change the SIMSWAPMON command to SWAP or SWAPDUMP and rerun the GENSWAP to generate the SWAP jobs.

- ❖ Use GENSWAP to generate and submit a SIMSWAP job for the desired set of source volumes. Set up all of the GENSWAP parameters as you want them for the following SIMSWAPMON, SWAP, or SWAPDUMP jobs in order to get an exact simulation.The PASJOB input for SIMSWAP contains model JCL for the one LPAR where SIMSWAP runs.
  - The listings from the SIMSWAP jobs show all of the LPARs that are connected to the source volumes. Look for the FDR233 message with the text "**FOLLOWING LPARS NEED MONITOR**" in the SYSPRINT at the end of a successful SIMSWAP to get a summary of all the LPARs that need MONITOR tasks.
  - Prepare for the SIMSWAPMON job and the SWAP or SWAPDUMP job by adding to the PASJOB input a monitor job for each system that is connected to the source volumes that you obtain from the output of the SIMSWAP listing.
- ❖ Use GENSWAP to generate and submit a SIMSWAPMON job, or a set of them. Set up parameters for the SWAP or SWAPDUMP, in order to get an exact simulation. Each SIMSWAPMON job generates and submits monitor jobs to the other systems.
- ❖ Use GENSWAP to generate and submit a SWAP/SWAPDUMP job, or a set of them. Change the command SIMSWAPMON to SWAP or SWAPDUMP, and do not change anything else. Each SWAP or SWAPDUMP job generates and submits monitor jobs to the other systems.

FIRST THREE EXAMPLES

The first three examples are a set that illustrates the recommended sequence of operations for GENSWAP:

- Example A: GENSWAP SIMSWAP
- Example B: GENSWAP SIMSWAPMON
- Example C: GENSWAP SWAP

The scenario for this set of examples is that the installation is replacing a control unit, and is going to use FDRPAS to SWAP all of the volumes in the old control unit, a little more than 3,000 volumes, to the new control unit.

On the GENSWAP command, `MAXGEN=200` causes a separate job to be generated for each 200 volumes. Although a SWAP or SWAPDUMP job can include up to 9999 volumes, a number like 200 makes the jobs more manageable and the user can put a hold on running jobs if needed. Also, there is less overhead in the monitors with fewer volumes. Following the guideline to keep the operands the same on the GENSWAP commands throughout the process, the user specifies `MAXGEN=200` for the SIMSWAP and SIMSWAPMON as well as the SWAP. In this example, 16 jobs are generated, with the last job containing the remainder after dividing by 200.

The MOUNT statements for GENSWAP specify the desired set of source volumes and stay the same throughout the set. In these examples, there is one MOUNT statement, and it selects all of the volumes in the control unit that is being replaced.

The number of generated SIMSWAP/SIMSWAPMON/SWAP jobs that run concurrently is controlled by the number of initiators that are active for the job class that is specified on the JOB cards (`CLASS=1` in these examples). The number of volumes that are processed concurrently by each SWAP job is controlled by the `MAXTASKS` operand (`MAXTASKS=10`). To increase the total number of volumes that are processed concurrently, increase the `MAXTASKS` value (up to 64) or increase the number of initiators. The number of initiators that are active for the job class that is specified on the JOB cards for the monitor jobs (`CLASS=2` in these examples) must be at least twice as many as for the class for the SIMSWAPMON or SWAP jobs. This is so that the next monitor can start while the previous one is terminating; otherwise, the next monitor might not be ready to respond in time. For example, if there are 3 initiators for the SWAP jobs, there must be 6 initiators, on each monitor system, for the monitor jobs.

NOTE: Following the guideline to keep the operands the same throughout the process, `MAXTASKS=10` is specified in all three examples, although it is ignored by SIMSWAP and SIMSWAPMON. SIMSWAP always processes one volume at a time, and SIMSWAPMON always processes 32 volumes at a time.

`LARGERSIZE=OK` is specified on the SIMSWAP, SIMSWAPMON, and SWAP commands, because some of the target volumes are larger in size than their corresponding source volumes (target volumes must always be as large as, or larger than, their source volumes). `LARGERSIZE=` does not need to be coded if all the volumes are the same size.

EXAMPLE A: GENSWAP SIMSWAP

Generate an FDRPAS SIMSWAP job stream in preparation for a SWAP. See the introduction under "First Three Examples" in Section 305.7. SIMSWAP should be run on one of the LPARs where the SWAP main task will run.

SIMSWAP verifies that the source volumes are online and that the target units exist and are offline. By examining the listing from SIMSWAP, you can confirm that all of the expected systems are connected to the source volumes, and that no unexpected systems are connected to the source volumes. The FDR233 message with the text "**FOLLOWING LPARS NEED MONITOR**" at the end of a successful SIMSWAP job provides a summary of all the unique LPARs that need MONITOR tasks. The end of the SYSPRINT output looks similar to:

```
…
FDR233  FOLLOWING LPARS NEED MONITOR
FDR233  CPU WITH (SERIAL# 0112342818)
FDR233  CPU WITH (SERIAL# 0212342818)
FDR233  CPU WITH (SERIAL# 0912342818)
FDR233  CPU WITH (SERIAL# 0512342818)
FDR233  CPU WITH (SERIAL# 0612342818)
FDR999   FDR SUCCESSFULLY COMPLETED
```

The PASJOB DD contains the model JCL for the SIMSWAP job. The SIMSWAP command includes operands that only apply to the subsequent SIMSWAPMON and SWAP jobs, in order to have these operands syntax checked and to minimize changes in later stages. The CHECKTARGET operand is not used by SIMSWAP, but is specified here as it will be used in subsequent jobs utilizing this model.

```
//GENSWAP  EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN  DD *
  GENSWAP  TYPE=FULL
  MOUNT    VOL=*,SWAPUNIT=5***,CUID=43928
/*
//PAS.PASJOB DD DATA,DLM=SS
//*SWAPJOB
//PASSS    JOB (ACCT),'STGADM',CLASS=1,MSGCLASS=X,
//            MSGLEVEL=(1,1),NOTIFY=userid
//SIMSWAP  EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN  DD *
  SIMSWAP  TYPE=FULL,CHECKSOURCE=YES,CHECKTARGET=YES,
           MAXTASKS=10,ALLOWPAV=YES,LARGERSIZE=OK
  MOUNT    VOL=&&&&&&,SWAPUNIT=&&&&
/*
SS                  <- End of outer PASJOB template
```

The INTRDR generated SIMSWAP job stream is similar to:

```
//PASSS    JOB (ACCT),'STGADM',CLASS=1,MSGCLASS=X,
//            MSGLEVEL=(1,1),NOTIFY=userid
//SIMSWAP  EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN  DD *
  SIMSWAP  TYPE=FULL,CHECKSOURCE=YES,CHECKTARGET=YES,
           MAXTASKS=10,ALLOWPAV=YES,LARGERSIZE=OK
  MOUNT    VOL=SYTS39,SWAPUNIT=5***                        <------1
  MOUNT    VOL=SYTS41,SWAPUNIT=5***                        <------2
  …
  MOUNT    VOL=CIC103,SWAPUNIT=5***                        <---3003
  MOUNT    VOL=SYSLB3,SWAPUNIT=5***                        <---3004
/*
```

When the SIMSWAP job is submitted, it produces messages for each volume, similar to:

```
CPK320I  COMPAKTOR OPTIONS IN EFFECT -- CHECKSOURCE-------------APWPM3
CPK512I  FORMAT 4 INDICATES VOLUME OCCUPIES     32760 CYLINDERS
CPK529I  INDEXED VTOC EXISTS ON VOLUME=APWPM3
CPK536I  VOLUME=APWPM3 IS A SYSTEM MANAGED VOLUME
VOLUME SUMMARY --- DEVICE TYPE ..............................  3390-27
                   NO. OF TRACKS ..............................491400
                   NO. OF CYLINDERS ........................... 32760
                   NO. OF ALLOCATED TRACKS .....................334566
                   PERCENTAGE OF VOLUME IN USE .................    68
CPK600I  COMPAKTOR SUCCESSFULLY COMPLETED CHECKSOURCE RUN ------ VOLUME=APWPM3
FDR233  CPU WITH (SERIAL# 0212342818) IS ATTACHED TO VOL=APWPM3              - EMC 2105F20 TO EMC 2107921
FDR233  CPU WITH (SERIAL# 0112342818) IS ATTACHED TO VOL=APWPM3              - EMC 2105F20 TO EMC 2107921
FDR233  CPU WITH (SERIAL# 0912342818) IS ATTACHED TO VOL=APWPM3              - EMC 2105F20 TO EMC 2107921
FDRW66  SWAP OF VOL=APWPM3 TO UNIT=5014 NEEDS TO BE STARTED ON   3 SYSTEMS
```

EXAMPLE B: GENSWAP SIMSWAPMON

Generate an FDRPAS SIMSWAPMON job stream in preparation for a SWAP. See the introduction under "First Three Examples" in Section 305.7.

- Review the output listings from the SIMSWAP jobs to make sure that all of the expected source volumes were found. We are expecting 3004 volumes generated. A sequence number is placed on every generated line, so this numeric value can be used to ensure that 3004 volumes were selected. If there are any differences in the number of volumes selected or other errors, correct them and rerun the SIMSWAP job.
- Change the SIMSWAP command to SIMSWAPMON.
- Set up the inner PASJOB DD, within the JCL for SIMSWAPMON. The inner PASJOB DD contains the model JCL for the monitor jobs that SIMSWAPMON (and later, SWAP or SWAPDUMP) submits to the other systems. Include a monitor job for each system shown in an FDR233 message. A SIMSWAPMON/SWAP/SWAPDUMP job automatically bypasses submitting a monitor to the LPAR on which it is running. The LPAR where SIMSWAP ran is shown in the *first* FDR233 message.
  - Each monitor job starts with a `//*CPUID=` statement identifying the LPAR where it runs.
  - After that comes the JOB statement. The job name must include one or more characters that make it unique (i.e. different from the job names for the SIMSWAPMON job and the other monitors); we suggest characters related to the system on which the monitor runs. If more than one SIMSWAPMON job (and later, the SWAP or SWAPDUMP job) is being generated, the job name for each monitor must also include one, two, or three ampersands (&). `CLASS=2` on the JOB card specifies a job class with two or more initiators, so that a monitor job can start while the previous monitor job is terminating.
  - Include the routing information that causes the monitor to run on the desired system (see Section 305.6 "GENSWAP Model Definition"). This example uses `/*ROUTE XEQ`.
- The monitors specify `DURATION=2` so that they terminate automatically after 2 minutes of inactivity. SIMSWAPMON/SWAP/SWAPDUMP changes `DURATION=` to `STOPINAC=`, which tells the monitors to stop even sooner, as soon as all of the specified SWAPUNITs have been processed.
- Initially, the `//PAS.INTRDR DD` statement for GENSWAP directs the output to SYSOUT for viewing. When you are satisfied with the generated jobs, remove this statement, and GENSWAP defaults to submitting the generated jobs to the internal reader for execution.

```
//PASGS    JOB (ACCT),'STGADM',CLASS=A,MSGCLASS=X,
//             MSGLEVEL=(1,1),NOTIFY=userid
//GENSWAP  EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.INTRDR DD SYSOUT=*   <--- remove after satisfactory review
//PAS.SYSIN  DD *
  GENSWAP  TYPE=FULL,MAXGEN=200,VOLSORT=YES
  MOUNT    VOL=*,SWAPUNIT=5***,CUID=43928
/*
//PAS.PASJOB DD DATA,DLM=SS
//*SWAPJOB
//PASSM&&  JOB (ACCT),'STGADM',CLASS=1,MSGCLASS=X,
//             MSGLEVEL=(1,1),NOTIFY=userid
//SSWAPMON EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN DD *
  SIMSWAPMON  TYPE=FULL,CHECKSOURCE=YES,CHECKTARGET=YES,
              MAXTASKS=10,ALLOWPAV=YES,LARGERSIZE=OK
  MOUNT    VOL=&&&&&&,SWAPUNIT=&&&&
/*
//PAS.PASJOB DD DATA,DLM=MM
//*CPUID=0912342818
//PASMA&&  JOB (ACCT),'STGADM',CLASS=2,MSGCLASS=X,
//             MSGLEVEL=(1,1),NOTIFY=userid
/*ROUTE  XEQ   JESCPUA
//MONITORA EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN DD *
  MONITOR  TYPE=SWAP,ALLOWPAV=YES,DURATION=2
  MOUNT    SWAPUNIT=&&&&
/*
//*CPUID=0212342818
//PASMB&&  JOB (ACCT),'STGADM',CLASS=2,MSGCLASS=X,
//             MSGLEVEL=(1,1),NOTIFY=userid
/*ROUTE  XEQ   JESCPUB
//MONITORB EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN DD *
  MONITOR  TYPE=SWAP,ALLOWPAV=YES,DURATION=2
  MOUNT    SWAPUNIT=&&&&
/*
//*CPUID=0112342818
//PASMC&&  JOB (ACCT),'STGADM',CLASS=2,MSGCLASS=X,
//             MSGLEVEL=(1,1),NOTIFY=userid
/*ROUTE  XEQ   JESCPUC
//MONITORC EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN DD *
  MONITOR  TYPE=SWAP,ALLOWPAV=YES,DURATION=2
  MOUNT    SWAPUNIT=&&&&
/*
MM                  <- End of inner MONITOR templates
SS                  <- End of outer PASJOB template
```

The INTRDR generated SIMSWAPMON job stream is similar to:

```
//PASSM01  JOB (ACCT),'STGADM',CLASS=1,MSGCLASS=X,
//             MSGLEVEL=(1,1),NOTIFY=userid
//SSWAPMON EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN DD *
  SIMSWAPMON  TYPE=FULL,CHECKSOURCE=YES,CHECKTARGET=YES,
              MAXTASKS=10,ALLOWPAV=YES,LARGERSIZE=OK
  MOUNT    VOL=SYTS39,SWAPUNIT=5***                      <------1
  MOUNT    VOL=SYTS41,SWAPUNIT=5***                      <------2
  …
  MOUNT    VOL=APPM01,SWAPUNIT=5***                      <----200
/*
//PAS.PASJOB DD DATA,DLM=MM
//*CPUID=0912342818,GENSWAP
//PASMA01  JOB (ACCT),'STGADM',CLASS=2,MSGCLASS=X,
//             MSGLEVEL=(1,1),NOTIFY=userid
/*ROUTE  XEQ   JESCPUA
//MONITORC EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN DD *
  MONITOR  TYPE=SWAP,DURATION=2
  MOUNT    SWAPUNIT=&&&&
/*
…
//*CPUID=0112342818,GENSWAP
//PASMC01  JOB (ACCT),'STGADM',CLASS=2,MSGCLASS=X,
//             MSGLEVEL=(1,1),NOTIFY=userid
/*ROUTE  XEQ   JESCPUC
//MONITORA EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN DD *
  MONITOR  TYPE=SWAP,DURATION=2
  MOUNT    SWAPUNIT=&&&&
/*
MM                  <- End of inner MONITOR templates
//PASSM02  JOB (ACCT),'STGADM',CLASS=1,MSGCLASS=X,
//             MSGLEVEL=(1,1),NOTIFY=userid
//SSWAPMON EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN DD *
  SIMSWAPMON  TYPE=FULL,CHECKSOURCE=YES,CHECKTARGET=YES,
              MAXTASKS=10,ALLOWPAV=YES,LARGERSIZE=OK
  MOUNT    VOL=SYTS16,SWAPUNIT=5***                      <----201
  MOUNT    VOL=APPM73,SWAPUNIT=5***                      <----202
  …
  MOUNT    VOL=VTS015,SWAPUNIT=5***                      <----400
/*
//PAS.PASJOB DD DATA,DLM=MM
//*CPUID=0912342818,GENSWAP
//PASMA02  JOB (ACCT),'STGADM',CLASS=2,MSGCLASS=X,
//             MSGLEVEL=(1,1),NOTIFY=userid
/*ROUTE  XEQ   JESCPUA
//MONITORA EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN DD *
                              (1 of 2)
```

```
  MONITOR  TYPE=SWAP,DURATION=2
  MOUNT    SWAPUNIT=&&&&
/*
…
//*CPUID=0112342818,GENSWAP
//PASMC02  JOB (ACCT),'STGADM',CLASS=2,MSGCLASS=X,
//             MSGLEVEL=(1,1),NOTIFY=userid
/*ROUTE  XEQ   JESCPUC
//MONITORC EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN DD *
  MONITOR  TYPE=SWAP,DURATION=2
  MOUNT    SWAPUNIT=&&&&
/*
MM                  <- End of inner MONITOR templates
…
//PASSM16  JOB (ACCT),'STGADM',CLASS=1,MSGCLASS=X,
//             MSGLEVEL=(1,1),NOTIFY=userid
//SSWAPMON EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN  DD *
  SIMSWAPMON  TYPE=FULL,CHECKSOURCE=YES,CHECKTARGET=YES,
              MAXTASKS=10,ALLOWPAV=YES,LARGERSIZE=OK
  MOUNT    VOL=CICS41,SWAPUNIT=5***                      <---3001
  MOUNT    VOL=CICS98,SWAPUNIT=5***                      <---3002
  MOUNT    VOL=CIC103,SWAPUNIT=5***                      <---3003
  MOUNT    VOL=SYSLB3,SWAPUNIT=5***                      <---3004
/*
//PAS.PASJOB DD DATA,DLM=MM
//*CPUID=0912342818,GENSWAP
//PASMA16  JOB (ACCT),'STGADM',CLASS=2,MSGCLASS=X,
//             MSGLEVEL=(1,1),NOTIFY=userid
/*ROUTE  XEQ   JESCPUA
//MONITORA EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN  DD *
  MONITOR  TYPE=SWAP,DURATION=2
  MOUNT    SWAPUNIT=&&&&
/*
…
//*CPUID=0112342818,GENSWAP
//PASMC16  JOB (ACCT),'STGADM',CLASS=2,MSGCLASS=X,
//             MSGLEVEL=(1,1),NOTIFY=userid
/*ROUTE  XEQ   JESCPUC
//MONITORC EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN DD *
  MONITOR  TYPE=SWAP,DURATION=2
  MOUNT    SWAPUNIT=&&&&
/*
MM                  <- End of inner MONITOR templates
```

(2 of 2)

EXAMPLE C: GENSWAP SWAP

Generate an FDRPAS SWAP job stream. See the introduction under "First Three Examples" in Section 305.7.

- Review the output listings from the SIMSWAPMON jobs. Correct any problems. Rerun the SIMSWAPMON jobs until they come out right. If you need to change the MOUNT statements, go back to the SIMSWAP jobs.
- Change the SIMSWAPMON command to SWAP.
- At this stage, no other changes are needed.
- Initially, the `//PAS.INTRDR DD` statement for GENSWAP directs the output to SYSOUT for viewing. When you are satisfied with the generated jobs, remove this statement, and GENSWAP defaults to submitting the generated SWAP jobs to the internal reader for execution.

The generated SWAP job stream is not shown here because it is exactly the same as the generated job stream in Example B: GENSWAP SIMSWAPMON, except for saying SWAP instead of SIMSWAPMON.

```
//PASGS    JOB (ACCT),'STGADM',CLASS=A,MSGCLASS=X,
//             MSGLEVEL=(1,1),NOTIFY=userid
//GENSWAP  EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.INTRDR DD SYSOUT=*   <--- remove after satisfactory review
//PAS.SYSIN  DD *
  GENSWAP  TYPE=FULL,MAXGEN=200,VOLSORT=YES
  MOUNT    VOL=*,SWAPUNIT=5***,CUID=43928
/*
//PAS.PASJOB DD DATA,DLM=SS
//*SWAPJOB
//PASSW&&  JOB (ACCT),'STGADM',CLASS=1,MSGCLASS=X,
//             MSGLEVEL=(1,1),NOTIFY=userid
//SSWAPMON EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN DD *
  SWAP     TYPE=FULL,CHECKSOURCE=YES,CHECKTARGET=YES,
              MAXTASKS=10,ALLOWPAV=YES,LARGERSIZE=OK
  MOUNT    VOL=&&&&&&,SWAPUNIT=&&&&
/*
//PAS.PASJOB DD DATA,DLM=MM
//*CPUID=0912342818
//PASMA&&  JOB (ACCT),'STGADM',CLASS=2,MSGCLASS=X,
//             MSGLEVEL=(1,1),NOTIFY=userid
/*ROUTE  XEQ   JESCPUC
//MONITORA EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN DD *
  MONITOR  TYPE=SWAP,ALLOWPAV=YES,DURATION=2
  MOUNT    SWAPUNIT=&&&&
/*
…
//*CPUID=0112342818
//PASMC&&  JOB (ACCT),'STGADM',CLASS=2,MSGCLASS=X,
//             MSGLEVEL=(1,1),NOTIFY=userid
/*ROUTE  XEQ   JESCPUA
//MONITORC EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN DD *
  MONITOR  TYPE=SWAP,ALLOWPAV=YES,DURATION=2
  MOUNT    SWAPUNIT=&&&&
/*
MM                  <- End of inner MONITOR templates
SS                  <- End of outer PASJOB template
```

GENERATE THE JCL AND CONTROL CARDS TO SWAP A NUMBER OF SSIDS IN ONE SEQUENTIAL JOB EXAMPLE

Generate a SWAP task to run on one LPAR and the MONITOR tasks for all the LPARs (including the LPAR where the SWAP task runs) to swap all the volumes in a SSID. The volumes in SSIDs 3000, 3001, and 3002 are swapped to corresponding units with an '80', '85', and '47' as the first digits in the UCB address. The SWAP job is set to process 32 volumes concurrently. An override of the `//PAS.INTRDR DD` statement directs the output to SYSOUT for viewing instead of submitting it directly to the internal reader for execution. Once you are satisfied with the job that is generated, remove the `//PAS.INTRDR DD` statement to allow the generated job to be submitted to the system for execution.

```
//PASGS    JOB (ACCT),'STGADM',CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
//GENSWAP  EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.INTRDR DD SYSOUT=*
//PAS.SYSIN  DD *
  GENSWAP  TYPE=FULL,MAXGEN=256,VOLSORT=YES
  MOUNT    VOL=*,SWAPUNIT=80**,SSID=3000
  MOUNT    VOL=*,SWAPUNIT=85**,SSID=3001
  MOUNT    VOL=*,SWAPUNIT=47**,SSID=3002
/*
//PAS.PASJOB DD DATA,DLM=SS
//*SWAPJOB
//PASSW&&  JOB (ACCT),'STGADM',CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
//SWAPTASK EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN DD *
  SWAP     TYPE=FULL,MAXTASKS=32,LARGERSIZE=OK
  MOUNT    VOL=&&&&&&,SWAPUNIT=&&&&
/*
//PAS.PASJOB DD DATA,DLM=MM
//*CPUID=0912342818
//PASMA&&  JOB (ACCT),'STGADM',CLASS=2,MSGCLASS=X,MSGLEVEL=(1,1)
/*ROUTE    XEQ JESCPUA
//MONITORA EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN DD *
  MONITOR  TYPE=SWAP,DURATION=2
  MOUNT    SWAPUNIT=&&&&
/*
//*CPUID=0212342818
//PASMB&&  JOB (ACCT),'STGADM',CLASS=2,MSGCLASS=X,MSGLEVEL=(1,1)
/*ROUTE    XEQ JESCPUB
//MONITORB EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN DD *
  MONITOR  TYPE=SWAP,DURATION=2
  MOUNT    SWAPUNIT=&&&&
/*
…
MM                  <- End of inner MONITOR templates
SS                  <- End of outer PASJOB template
```

NOTE 1: Generate MONITOR tasks for all the LPARs (including the LPAR where the SWAP task runs). The SWAP task only submits the MONITOR tasks where they are needed. Having all the MONITOR tasks provides flexibility and portability in that the SWAP job can be run on another LPAR and all the necessary MONITOR tasks would then be submitted.

NOTE 2: This job gives the flexibility to move sets of SSIDs when a sufficient number of target volumes are unavailable to move all the desired volumes by specifying a high-level digit for the SWAPUNIT value. The selection of the target volumes can be controlled by specifying the first two digits of the SWAPUNIT causing the job to process only 256 volumes matching the unit addresses.

The INTRDR generated output is similar to:

```
//PASSW01  JOB (ACCT),'STGADM',CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
//SWAPTASK EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN DD *
  SWAP     TYPE=FULL,MAXTASKS=32,LARGERSIZE=OK
  MOUNT    VOL=DB2EU1,SWAPUNIT=80**                      <------1
  MOUNT    VOL=SYM000,SWAPUNIT=85**                      <------2
  MOUNT    VOL=SYM015,SWAPUNIT=47**                      <------3
  MOUNT    VOL=SYM019,SWAPUNIT=80**                      <------4
  MOUNT    VOL=SYM009,SWAPUNIT=85**                      <------5
  MOUNT    VOL=SYM003,SWAPUNIT=47**                      <------6
…
  MOUNT    VOL=DV023A,SWAPUNIT=47**                      <----255
  MOUNT    VOL=PRCE03,SWAPUNIT=80**                      <----256
/*
//PAS.PASJOB DD DATA,DLM=MM
//*CPUID=0912342818,GENSWAP
//PASMA01  JOB (ACCT),'STGADM',CLASS=2,MSGCLASS=X,MSGLEVEL=(1,1)
/*ROUTE    XEQ JESCPUA
//MONITORA EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN  DD *
  MONITOR  TYPE=SWAP,DURATION=2
  MOUNT    SWAPUNIT=&&&&
/*
//*CPUID=0212342818,GENSWAP
//PASMB01  JOB (ACCT),'STGADM',CLASS=2,MSGCLASS=X,MSGLEVEL=(1,1)
/*ROUTE    XEQ JESCPUB
//MONITORB EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN  DD *
  MONITOR  TYPE=SWAP,DURATION=2
  MOUNT    SWAPUNIT=&&&&
/*
//*CPUID=0112342818
//PASMC01  JOB (ACCT),'STGADM',CLASS=2,MSGCLASS=X,MSGLEVEL=(1,1)
/*ROUTE    XEQ JESCPUC
//MONITORC EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN  DD *
  MONITOR  TYPE=SWAP,DURATION=2
  MOUNT    SWAPUNIT=&&&&
/*
MM                  <- End of inner MONITOR templates
//PASSW02  JOB (ACCT),'STGADM',CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
//SWAPTASK EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN DD *
  SWAP     TYPE=FULL,MAXTASKS=32,LARGERSIZE=OK
  MOUNT    VOL=SYM007,SWAPUNIT=85**                      <----257
  MOUNT    VOL=SYM018,SWAPUNIT=47**                      <----258
…
```

GENERATE THE JCL AND CONTROL CARDS TO SWAP AN ENTIRE CUID IN MULTIPLE SEQUENTIAL JOBS EXAMPLE

Generate SWAP jobs to run on one LPAR and all the MONITOR tasks for all the LPARs to swap all the volumes in a CUID (excluding the volumes containing the local PAGE data sets and volumes containing sysplex couple data sets). The volumes in CUID=65950 are swapped with 1024 volumes specified per job. This is controlled by the MAXGEN=1024 operand. CLASS=S is used in the generated SWAP job as this is defined to the site as an initiator for running jobs sequentially. The SWAPID= operand is used to generate sequential numbers on the generated SWAP jobs and associated MONITOR jobs.

This can be used to copy an entire Control Unit (CUID) with a maximum of 4096 unit addresses to a target CUID that is configured similar to the source.

NOTE: The MAXGEN=1024 operand causes GENSWAP to generate up to 1024 volumes in a single job. When this limit is reached, a new SWAP job is created.

```
//GENSWAP  EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.INTRDR DD SYSOUT=*
//PAS.SYSIN  DD *
  GENSWAP  TYPE=FULL,MAXGEN=1024,VOLSORT=YES
  EXCLUDE  VOL=PAGE*           EXCLUDE ALL LOCAL PAGE VOLUMES
  EXCLUDE  VOL=SYXCF*          EXCLUDE VOLS WITH COUPLE DATA SET
  MOUNT    VOL=*,SWAPUNIT=5***,CUID=65950
/*
//PAS.PASJOB DD DATA,DLM=SS
//*SWAPJOB
//PASSW&   JOB (ACCT),'STGADM',CLASS=S,MSGCLASS=X,MSGLEVEL=(1,1)
//SWAPTASK EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN  DD *
  SWAP     TYPE=FULL,MAXTASKS=32,LARGERSIZE=OK,SWAPID=&
  MOUNT    VOL=&&&&&&,SWAPUNIT=&&&&
/*
//PAS.PASJOB DD DATA,DLM=MM
//*CPUID=0912342818
//PASMA&   JOB (ACCT),'STGADM',CLASS=2,MSGCLASS=X,MSGLEVEL=(1,1)
/*ROUTE    XEQ JESCPUA
//MONITORA EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN DD *
  MONITOR  TYPE=SWAP,DURATION=2,SWAPID=&
  MOUNT    SWAPUNIT=&&&&
/*
//*CPUID=0212342818
//PASMB&   JOB (ACCT),'STGADM',CLASS=2,MSGCLASS=X,MSGLEVEL=(1,1)
/*ROUTE    XEQ JESCPUB
//MONITORB EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN DD *
  MONITOR  TYPE=SWAP,DURATION=2,SWAPID=&
  MOUNT    SWAPUNIT=&&&&
/*
//*CPUID=0112342818
//PASMC&   JOB (ACCT),'STGADM',CLASS=2,MSGCLASS=X,MSGLEVEL=(1,1)
/*ROUTE    XEQ JESCPUC
//MONITORC EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN DD *
  MONITOR  TYPE=SWAP,DURATION=2,SWAPID=&
  MOUNT    SWAPUNIT=&&&&
/*
MM                  <- End of inner MONITOR templates
SS                  <- End of outer PASJOB template
```

The INTRDR generated output is similar to:

```
//PASSW1   JOB (ACCT),'STGADM',CLASS=S,MSGCLASS=X,MSGLEVEL=(1,1)
//SWAPTASK EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN  DD *
  SWAP     TYPE=FULL,MAXTASKS=32,LARGERSIZE=OK,SWAPID=1
  MOUNT    VOL=DV17FF,SWAPUNIT=5***                      <------1
  MOUNT    VOL=DV17F6,SWAPUNIT=5***                      <------2
…
  MOUNT    VOL=PRCH01,SWAPUNIT=5***                      <---1023
  MOUNT    VOL=HI17D1,SWAPUNIT=5***                      <---1024
/*
//PAS.PASJOB DD DATA,DLM=MM
//*CPUID=0912342818,GENSWAP
//PASMA1   JOB (ACCT),'STGADM',CLASS=2,MSGCLASS=X,MSGLEVEL=(1,1)
/*ROUTE    XEQ JESCPUA
//MONITORA EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN DD *
  MONITOR  TYPE=SWAP,DURATION=2,SWAPID=&
  MOUNT    SWAPUNIT=&&&&
/*
//*CPUID=0212342818,GENSWAP
//PASMB1   JOB (ACCT),'STGADM',CLASS=2,MSGCLASS=X,MSGLEVEL=(1,1)
/*ROUTE    XEQ JESCPUB
//MONITORB EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN  DD *
  MONITOR  TYPE=SWAP,DURATION=2,SWAPID=&
  MOUNT    SWAPUNIT=&&&&
/*
//*CPUID=0112342818,GENSWAP
//PASMC1   JOB (ACCT),'STGADM',CLASS=2,MSGCLASS=X,MSGLEVEL=(1,1)
/*ROUTE    XEQ JESCPUC
//MONITORC EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN  DD *
  MONITOR  TYPE=SWAP,DURATION=2,SWAPID=&
  MOUNT    SWAPUNIT=&&&&
/*
MM                  <- End of inner MONITOR templates
//PASSW2   JOB (ACCT),'STGADM',CLASS=S,MSGCLASS=X,MSGLEVEL=(1,1)
//SWAPTASK EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN  DD *
  SWAP     TYPE=FULL,MAXTASKS=32,LARGERSIZE=OK,SWAPID=2
  MOUNT    VOL=HI17C6,SWAPUNIT=5***                      <---1025
  MOUNT    VOL=HI17C8,SWAPUNIT=5***                      <---1026
…
  MOUNT    VOL=DV186D,SWAPUNIT=5***                      <---2047
  MOUNT    VOL=HI1859,SWAPUNIT=5***                      <---2048
/*
//PAS.PASJOB DD DATA,DLM=MM
//*CPUID=0912342818,GENSWAP
//PASMA2   JOB (ACCT),'STGADM',CLASS=2,MSGCLASS=X,MSGLEVEL=(1,1)
/*ROUTE    XEQ JESCPUA
                              (1 of 2)
```

```
//MONITORA EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN DD *
  MONITOR  TYPE=SWAP,DURATION=2,SWAPID=&
  MOUNT    SWAPUNIT=&&&&
/*
//*CPUID=0212342818,GENSWAP
//PASMB2   JOB (ACCT),'STGADM',CLASS=2,MSGCLASS=X,MSGLEVEL=(1,1)
/*ROUTE    XEQ JESCPUB
//MONITORB EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN  DD *
  MONITOR  TYPE=SWAP,DURATION=2,SWAPID=&
  MOUNT    SWAPUNIT=&&&&
/*
//*CPUID=0112342818,GENSWAP
//PASMC2   JOB (ACCT),'STGADM',CLASS=2,MSGCLASS=X,MSGLEVEL=(1,1)
/*ROUTE    XEQ JESCPUC
//MONITORC EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN  DD *
  MONITOR  TYPE=SWAP,DURATION=2,SWAPID=&
  MOUNT    SWAPUNIT=&&&&
/*
MM                  <- End of inner MONITOR templates
//PASSW3   JOB (ACCT),'STGADM',CLASS=S,MSGCLASS=X,MSGLEVEL=(1,1)
//SWAPTASK EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN  DD *
  SWAP     TYPE=FULL,MAXTASKS=32,LARGERSIZE=OK,SWAPID=3
  MOUNT    VOL=DV17F3,SWAPUNIT=5***                      <---2049
  MOUNT    VOL=DV17F8,SWAPUNIT=5***                      <---2050
…
                              (2 of 2)
```

GENERATE THE JCL AND CONTROL CARDS TO SWAP THE JES, PAGE, AND COUPLE VOLUMES EXAMPLE

FDRPAS supports swapping of JES, PAGE, and COUPLE volumes with other volumes. However, if you wish to control when these volumes are swapped, you can swap these volumes separately from the other volumes.

Generate SWAP jobs to run on one LPAR of a three LPAR system and the associated MONITOR tasks to run on the other two LPARs to swap the volumes containing the JES, local page data sets, and sysplex couple data sets that are in CUID=65950. `CLASS=S` is used in the generated SWAP job as this is defined to the site as an initiator for running jobs sequentially. An override of the `//PAS.INTRDR DD` statement directs the output to SYSOUT for viewing instead of submitting it directly to the internal reader for execution. Once you are satisfied with the job that is generated, remove the `//PAS.INTRDR DD` statement to allow the generated job to be submitted to the system for execution. This job processes 1 volume at a time.

```
//GENSWAP  EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.INTRDR DD SYSOUT=*
//PAS.SYSIN  DD *
  GENSWAP  TYPE=FULL,MAXGEN=1000,VOLSORT=YES
  MOUNT    VOL=JES*,SWAPUNIT=5***,CUID=65950
  MOUNT    VOL=PAGE*,SWAPUNIT=5***,CUID=65950
  MOUNT    VOL=SYXCF*,SWAPUNIT=5***,CUID=65950
/*
//PAS.PASJOB DD DATA,DLM=SS
//*SWAPJOB
//PASSW&&& JOB (ACCT),'STGADM',MSGCLASS=X,MSGLEVEL=(1,1),
// CLASS=& A,B,C,D,E
//SWAPTASK EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN  DD *
  SWAP     TYPE=FULL,MAXTASKS=1,LARGERSIZE=OK
  MOUNT    VOL=&&&&&&,SWAPUNIT=&&&&
/*
//PAS.PASJOB DD DATA,DLM=MM
//*CPUID=0912342818
//PASMA&&&  JOB (ACCT),'STGADM',MSGCLASS=X,MSGLEVEL=(1,1),
// CLASS=& A,B,C,D,E
/*ROUTE    XEQ JESCPUA
//MONITORA EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN DD *
  MONITOR  TYPE=SWAP,DURATION=2
  MOUNT    SWAPUNIT=&&&&
/*
//*CPUID=0212342818
//PASMB&&&  JOB (ACCT),'STGADM',MSGCLASS=X,MSGLEVEL=(1,1),
// CLASS=& A,B,C,D,E
/*ROUTE    XEQ JESCPUB
//MONITORB EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN DD *
  MONITOR  TYPE=SWAP,DURATION=2
  MOUNT    SWAPUNIT=&&&&
/*
MM                  <- End of inner MONITOR templates
SS                  <- End of outer PASJOB template
```

NOTE: If you wish to run more jobs concurrently because of time constraints or other reasons, you can modify the `MAXGEN=1000` operand to `MAXGEN=1`. With `MAXGEN=1` specified, the number of concurrent jobs is only limited by the number of initiators that are able to run the SWAP jobs based on the specified classes. The `// CLASS=& A,B,C,D,E` operand on the JOB statement is not needed when processing volumes 1 at a time, but was added in the event you wish to run multiple jobs concurrently and utilize this operand.

The INTRDR generated output is similar to:

```
//PASSW001 JOB (ACCT),'STGADM',MSGCLASS=X,MSGLEVEL=(1,1),
// CLASS=A A,B,C,D,E
//SWAPTASK EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN  DD *
  SWAP     TYPE=FULL,MAXTASKS=1,LARGERSIZE=OK
  MOUNT    VOL=JESB03,SWAPUNIT=5***                      <------1
  MOUNT    VOL=JESC01,SWAPUNIT=5***                      <------2
  MOUNT    VOL=JESE00,SWAPUNIT=5***                      <------3
  MOUNT    VOL=JESB02,SWAPUNIT=5***                      <------4
  MOUNT    VOL=JESSAV,SWAPUNIT=5***                      <------5
  MOUNT    VOL=JESA01,SWAPUNIT=5***                      <------6
  MOUNT    VOL=JESB04,SWAPUNIT=5***                      <------7
  MOUNT    VOL=JESA02,SWAPUNIT=5***                      <------8
  MOUNT    VOL=JESB01,SWAPUNIT=5***                      <------9
  MOUNT    VOL=JESC02,SWAPUNIT=5***                      <-----10
  MOUNT    VOL=PAGEB1,SWAPUNIT=5***                      <-----11
  MOUNT    VOL=PAGE01,SWAPUNIT=5***                      <-----12
  MOUNT    VOL=PAGEE0,SWAPUNIT=5***                      <-----13
  MOUNT    VOL=PAGEA0,SWAPUNIT=5***                      <-----14
  MOUNT    VOL=PAGEC0,SWAPUNIT=5***                      <-----15
  MOUNT    VOL=PAGEB0,SWAPUNIT=5***                      <-----16
  MOUNT    VOL=PAGEB2,SWAPUNIT=5***                      <-----17
  MOUNT    VOL=SYXCF1,SWAPUNIT=5***                      <-----18
  MOUNT    VOL=SYXCF2,SWAPUNIT=5***                      <-----19
/*
//PAS.PASJOB DD DATA,DLM=MM
//*CPUID=0912342818,GENSWAP
//PASMA001 JOB (ACCT),'STGADM',MSGCLASS=X,MSGLEVEL=(1,1),
// CLASS=& A,B,C,D,E
/*ROUTE    XEQ JESCPUA
//MONITORA EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN DD *
  MONITOR  TYPE=SWAP,DURATION=2
  MOUNT    SWAPUNIT=&&&&
/*
//*CPUID=0212342818,GENSWAP
//PASMB001 JOB (ACCT),'STGADM',MSGCLASS=X,MSGLEVEL=(1,1),
// CLASS=& A,B,C,D,E
/*ROUTE    XEQ JESCPUB
//MONITORB EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN DD *
  MONITOR  TYPE=SWAP,DURATION=2
  MOUNT    SWAPUNIT=&&&&
/*
MM                  <- End of inner MONITOR templates
```

305.8 FDRPAS FOR LARGE SCALE SYNCHRONIZED MIGRATION

OVERVIEW AND DESCRIPTION

The typical way to run FDRPAS is to allow the SWAP or SWAPDUMP of each volume to complete as soon as the data is completely copied. This section deals with situations where it is necessary to complete all of the volumes at the same time and thousands of volumes are involved.

SWAP

One FDRPAS SWAP job can be used to migrate any number of DASD volumes. It can process up to 64 volumes at a time, swapping each volume to its target after the data is copied, and then starting the next volume. While this job is underway, some of the volumes are on the source devices and some are on the target devices. All of the volumes are available for use throughout the process. After the job finishes, each source device will be a point-in-time backup as of the moment that volume was swapped.

However, in some cases it is necessary that all of the volumes be swapped at the same time. These cases include:

- **HyperSwap** – to keep HyperSwap to the least amount of downtime. HyperSwap must be disabled during the actual UCB swap process, so it is important to complete all of the UCB swaps in as short a time as possible. Typically, there are thousands of volumes controlled by HyperSwap.
- **EMC AutoSwap** – to keep AutoSwap to the least amount of downtime. AutoSwap must be disabled during the actual UCB swap process, so it is important to complete all of the UCB swaps in as short a time as possible. Typically, there are thousands of volumes controlled by AutoSwap.
- **Consistency Groups** – to maintain consistency when migrating to hardware from a different manufacturer. Consistency cannot be maintained when the volumes are not all on hardware from the same manufacturer, so it is important to complete all of the UCB swaps to the new hardware in as short a time as possible.
- **Point-In-Time Backup** – Some installations may have a requirement that all of the volumes be swapped at the same time so that the entire set of source devices can serve as a consistent point-in-time backup. Then, if the new DASD subsystems fail, the installation can revert to the old subsystems.

  NOTE: The swaps are not exactly at the same instant. The installation must quiesce application I/O for a short time during the actual UCB swaps if it is desired that the source volumes be consistent in the sense of dependent writes.

The `CONFIRMSWAP=YES` operand is used to keep the volumes synchronized until you are ready to:

- Bring down HyperSwap or AutoSwap
- Switch the consistency group to the new hardware
- Form the consistent backup

SWAPDUMP

One FDRPAS SWAPDUMP job can be used to copy any number of DASD volumes. It can process up to 64 volumes at a time, splitting off each target volume after the data is copied, and then starting the next volume. After the job finishes, each target volume is a point-in-time image of its source volume as of the moment that volume was split off.

However, in some cases it is necessary or desirable that all of the volumes be split off at the same time. These cases include:

- **Duplicating a System** – at the same location. Typically, this is used to create a test environment or a copy for Disaster Recovery testing.
- **Creating a New System** – at a different location. In this case, the volumes typically are duplicated in a DASD subsystem at a remote site connected by high-speed channel extenders or extended length channels.

The `CONFIRMSPLIT=YES` option can be used to keep the volumes synchronized until you are ready to split off the duplicated/new LPAR, creating the duplicated/new volumes as a point-in-time image. Usually, you will want to quiesce your LPARs as far as possible (so that no unnecessary jobs or online systems are running) during the split to get a consistent image of the data for use on the new LPAR.

SWAP and SWAPDUMP

NOTE: In the rest of this section, "SWAP" is used to mean either SWAP or SWAPDUMP, and "CONFIRM" is used to mean either `CONFIRMSWAP=YES` for SWAP or `CONFIRMSPLIT=YES` for SWAPDUMP, except where it is important to distinguish.

Unlike a SWAP without CONFIRM, where the swap of each volume is completed in a few minutes, SWAP with CONFIRM requires that the SWAP tasks of all the volumes involved continue to run until the point of the actual UCB swap (for SWAP) or the split off (for SWAPDUMP). Doing this for a large number of volumes requires many concurrent SWAP jobs, and special procedures. In FDRPAS V5.4/80 and above, FDRPAS provides facilities to simplify and automate the process of setting up all of the necessary SWAP and MONITOR jobs.

There are several considerations with running a large number of concurrent SWAP tasks:

- A given FDRPAS SWAP step can handle up to 64 volumes concurrently, if there is sufficient memory (see next bullet), so the SWAP tasks must be broken up into a number of jobs of 64 volumes or fewer in each job (the GENSWAP process handles this). You must have available initiators to run all of these jobs. It is possible to run the monitor jobs as started tasks so that they do not require initiators.
- Specifying `MAXTASKS=64` requires 10MB of below-the-line region size. Ensure the LPARs where the SWAP jobs run have this below-the-line region size available. If not, reduce the `MAXGEN=` and `MAXTASKS=` value by 8 for every MB less than 10MB in the private area. Reducing the `MAXGEN=` value creates more SWAP jobs to process the lower number of volumes processed per job.
- `MAXACTIVESWAPS=`*`nnn`* should be specified on the GENSWAP statement, so that no more than "*nnn*" volumes per LPAR are in the initial copy phase at the same time.
- If multiple LPARs are involved, MONITOR tasks are required.

  RECOMMENDATION: Run one MONITOR task on each LPAR corresponding to each SWAP job (the GENSWAP process handles this).
- FDRPAS does enqueues with major names of FDRPAS, FDRPASQ*, and FDRPASU* with SCOPE=SYSTEMS to indicate that swaps are in progress. The enqueues are used to detect duplicate swap requests and inhibit certain operations. It is required that these enqueues be propagated to all LPARs involved in the swap. If these enqueues are not propagated to some LPARs, FDRPAS may not be able to detect duplicate swap requests and the ISPF panels on LPARs running MONITOR tasks do not detect the swap in progress until

synchronization has completed on all LPARs. You should **NOT** convert the FDRPAS, FDRPASQ*, and FDRPASU* enqueues to SCOPE=SYSTEM. MIM users may need to add these major names to a MIM-list in order to propagate them. If you are submitting SWAP jobs to run on multiple LPARs, the LPARs running main SWAP jobs must be in the same PLEX (GRS or MIM).

- We recommend running concurrent SWAP jobs on no more than six LPARs.
- The LPARs on which SWAP jobs run must have enough CPU time available that the SWAP jobs can get CPU time when they need it. The SWAP jobs do not have to have a high dispatching priority, but the LPAR must not be capped or weighted or CPU-bound in a way that causes the SWAP jobs to be locked out for long periods.
- We recommend running with no more than 2500 volumes per LPAR in SWAP jobs, if enough LPARs are available. This is, if the number of volumes per LPAR would exceed 2500, it is preferable to add LPARs, if available.
- If you are submitting SWAP jobs to run on multiple LPARs, all of the volumes being swapped must be online on all of the LPARs running SWAP jobs. (All of the source volumes being swapped do not have to be online on LPARs that are only running monitors.)
- (*SWAPDUMP operation only.*) If performing a clone of an LPAR or creating a new LPAR, all target volumes should be marked as OFFLINE=YES in the IODF (I/O definition file) on the original LPAR since the SWAPDUMP marks all those volumes with the same volume serials as the online volumes on that LPAR. If an IPL is done before the duplicate volumes are disconnected and they are not marked offline, many operator messages are displayed and the wrong volumes might be varied online. Of course, these volumes must not be marked offline on the duplicated/new LPAR.

COMMON STORAGE REQUIREMENTS

FDRPAS requires some common memory (CSA and ECSA) on each LPAR for each active volume, so the requirements for running a large number of SWAP tasks can be significant. If the available CSA/ECSA is insufficient on one or more LPARs, FDRPAS errors or even system failures can occur. The details of FDRPAS CSA usage and how to check available CSA are in a separate document, but here is a brief summary:

You can display the current CSA and ECSA size and usage with this job:

```
//FDRDEBUG EXEC PGM=FDRDEBUG,PARM='FDRDEBUG SYS'
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//FDRDEBUG DD SYSOUT=*
```

This should be run on **EVERY** LPAR where the volumes involved are online and should be run during a busy time to get the maximum usage values.

There is a fixed requirement for below-the-line CSA. If LARGESWAP= is not specified, the requirement is 4 K (4096 bytes) for each 10 volumes (rounded up). For example, for 5000 volumes, the requirement for below-the-line CSA would be 500 x 4 K = 2000 KB = 2 MB. For a SWAP or SWAPDUMP of more than 2000 volumes, specify LARGESWAP= to reduce the CSA requirement. The LARGESWAP= value should be slightly larger than the number of volumes being swapped. For example, for 5000 volumes, specify LARGESWAP=6000. When LARGESWAP= specifies a value larger than the number of volumes being swapped, the requirement for below-the-line CSA is no more than 400 KB.

There is also a requirement for above-the-line ECSA. Each volume needs 1 K (1024 bytes) for the entire duration of the SWAP or SWAPDUMP. Each volume also needs a table of updated tracks. The size of this table is 2 times the number of data cylinders on the volume. Here are the sizes of the track table for some standard device sizes.

Standard Device	Number of Cylinders	# ECSA Bytes Required
3390-1	1,113	2226
3390-2	2,226	4452
3390-3	3,339	6678
3390-9	10,017	20,034
3390-27	32,760	65,520
3390-54	65,520	131,040

This table is usually needed only for a short time after any tracks have been updated, and only on the LPAR where the updates were done. It is difficult to estimate what proportion of the volumes will be using this table at a given time, but it may be reasonable to suppose that no more than 10% of the volumes being swapped will have had tracks updated within the last minute or so, on a given LPAR.

Example: For a SWAP of 5000 3390-9 volumes:

```
5000 volumes * 1 K constant requirement = .............5000 K
5000 * 10% = 500 * 20,034 bytes = ......................9782 K
          Total ....................................... 14,782 K
                                                    (or about 15 MB)
```

CSA and ECSA are also allocated dynamically by various system components; online and database systems are large users of CSA/ECSA. So the free CSA/ECSA values from the FDRDEBUG display must be comfortably above the FDRPAS requirements to allow for other usage. If the free values are inadequate **ON ANY LPAR**, you need to increase the CSA/ECSA sizes in PARMLIB and re-IPL that LPAR before attempting to swap a large number of volumes.

Active DASD I/Os also need some amount of ESQA acquired by IOS and EXCP, so if FDRPAS SWAP jobs are actively copying data for a large number of volumes, this IOS requirement could also total many KB. The SWAP procedure documented here does not allow more than 128 volumes to be in phase 1, the initial copy phase, at one time, so this should not be a problem.

GENERATE THE JCL AND CONTROL CARDS FOR A CONCURRENT SWAPDUMP OF 15,000 VOLUMES EXAMPLE

Generate SWAPDUMP jobs to run on six LPARs (JESCPUA - JESCPUF) and the associated MONITOR tasks to run on all ten of the attached LPARs (JESCPUA - JESCPUJ) for a SWAPDUMP of a total of 15,000 volumes. A special coding of the `/*ROUTE XEQ` statement is used to distribute the SWAPDUMP jobs across the six LPARs. An override of the `//PAS.INTRDR DD` statement directs the output to SYSOUT for viewing instead of submitting it directly to the internal reader for execution. Once you are satisfied with the jobs that are generated, remove the `//PAS.INTRDR DD` statement to allow the generated jobs to be submitted to the LPARs for execution. This example can be found in the JCL library installed with FDRPAS with member name PA30508A.

- The `MAXGEN=64` operand causes GENSWAP to generate up to 64 volumes in a single job. When this limit is reached, a new SWAPDUMP job is created. 40 initiators per LPAR are needed for the main SWAPDUMP tasks to process 2500 volumes per LPAR. $(2500/64 = 40)$.
- Specifying `MAXTASKS=64` requires 10MB of below-the-line region size. Ensure that the LPARs where the SWAPDUMP jobs run have this below-the-line region size available; if not, reduce the `MAXGEN=` and `MAXTASKS=` values by 8 for every MB less than 10 in the private area. Reducing the `MAXGEN=` value creates more SWAPDUMP jobs to process the lower number of volumes processed per job.
- `MAXACTIVESWAPS=10` is specified to allow a maximum of 60 SWAPDUMP tasks to be active at one time (10 on each of the 6 systems).
- `SWAPID=&&&` is specified on the SWAPDUMP and MONITOR commands to connect the MONITOR tasks with the associated SWAPDUMP jobs.
- `CPYVOLID=YES` is not documented in Section 310.2 "SWAP, SWAPDUMP, SIMSWAP, SIMSWAPMON Statements", but it is needed for this operation. SWAPDUMP normally leaves the volume label of the target volume in a state where the volume cannot be varied online. `CPYVOLID=YES` causes SWAPDUMP to leave the volume label in a normal state; but note that it duplicates the volume serial of the source volume.
- Remove `LARGERSIZE=OK` if the target volumes are the same sizes as their source volumes.
- If the SWAPDUMP job is to run for a long period of time before completing, it is recommended that `TIME=1440` (or `TIME=NOLIMIT`) be coded on the SWAPDUMP job step to avoid an S322 ABEND caused by a job exceeding the allowable CPU time.
- The monitors are run as Started Tasks (STCs) so they do not require initiators, except briefly when program FDREMCS issues the `START` commands.
- The *dsname* specified for the `IN=` operand on the `START` command for the MONITOR task must be the same as the *dsname* specified on the SYSUT2 DD statement.
- Ensure that the SIMSWAP and SIMSWAPMON jobs ran successfully and with the desired results before running the SWAPDUMP job.
- The special volumes (JES spool and checkpoint volumes and volumes containing sysplex coupling data sets) are included in this generated job stream. FDRPAS automatically delays the processing of these volumes until the end of the SWAPDUMP process to ensure they are swapped just before the confirm process.

```
//GENSWAP  EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.INTRDR DD SYSOUT=*
//PAS.SYSIN  DD *
  GENSWAP  TYPE=FULL,MAXGEN=64,VOLSORT=YES,SUBMITDELAY=15,
           MAXACTIVESWAPS=10,MAXCARDS=16000
  MOUNT    VOL=*,SWAPUNIT=5***,CUID=65950
  …
  MOUNT    VOL=*,SWAPUNIT=A***,CUID=22398
```

```
/*
//PAS.PASJOB DD DATA,DLM=SS
//*SWAPJOB
//PASSD&&& JOB (ACCT),'STGADM',MSGCLASS=X,MSGLEVEL=(1,1),CLASS=M
/*ROUTE XEQ & JESCPUA,JESCPUB,JESCPUC,JESCPUD,JESCPUE,JESCPUF
//SWAPDUMP EXEC PASPROC,LIB=fdrpas.loadlib,TIME=1440
//PAS.SYSIN  DD *
  SWAPDUMP TYPE=FULL,MAXTASKS=64,CONFIRMSPLIT=YES,CPYVOLID=YES,
           LARGESWAP=16000,SWAPID=&&&,ALLOWPAV=YES,LARGERSIZE=OK
  MOUNT    VOL=&&&&&&,SWAPUNIT=&&&&
/*
//PAS.PASJOB DD DATA,DLM=MM
//*CPUID=0912342818
//PASMA&&& JOB (ACCT),'STGADM',CLASS=M,MSGCLASS=X,MSGLEVEL=(1,1)
/*ROUTE    XEQ JESCPUA
//SUBMONA  EXEC PASPROC,PROG=FDREMCS,LIB=fdrpas.loadlib
//SYSUT1   DD *
  MONITOR  TYPE=SWAP,DURATION=2,SWAPID=&&&,
           LARGESWAP=16000,ALLOWPAV=YES
  MOUNT    SWAPUNIT=&&&&
/*
//SYSUT2   DD UNIT=temp_unit_name,
//            DSN=fdrpas.&JOBNAME.input.to.monitor,
//            DISP=(,CATLG),SPACE=(TRK,(1,1)),
//            DCB=BLKSIZE=3200
//SYSIN    DD *
 COMMAND=S PASPROC,JOBNAME=&JOBNAME,
         LIB=fdrpas.loadlib,
         IN=fdrpas.&JOBNAME.input.to.monitor,
         D='(OLD,DELETE),FREE=CLOSE'
/*
… (replicate the MONITOR tasks for all the LPARs)
//*CPUID=2912342818
//PASMJ&&& JOB (ACCT),'STGADM',CLASS=M,MSGCLASS=X,MSGLEVEL=(1,1)
/*ROUTE    XEQ JESCPUJ
//SUBMONJ  EXEC PASPROC,PROG=FDREMCS,LIB=fdrpas.loadlib
//SYSUT1   DD *
  MONITOR  TYPE=SWAP,DURATION=2,SWAPID=&&&,
           LARGESWAP=16000,ALLOWPAV=YES
  MOUNT    SWAPUNIT=&&&&
/*
//SYSUT2   DD UNIT=temp_unit_name,
//            DSN=fdrpas.&JOBNAME.input.to.monitor,
//            DISP=(,CATLG),SPACE=(TRK,(1,1)),
//            DCB=BLKSIZE=3200
//SYSIN    DD *
 COMMAND=S PASPROC,JOBNAME=&JOBNAME,
         LIB=fdrpas.loadlib,
         IN=fdrpas.&JOBNAME.input.to.monitor,
         D='(OLD,DELETE),FREE=CLOSE'
/*
MM                  <- End of inner MONITOR templates
SS                  <- End of outer PASJOB template
```

COMPLETING THE SWAP OR SWAPDUMP OPERATION

Once **ALL** the volumes have reached synchronization, the SWAP or SWAPDUMP operation can be completed. "Completed" means issuing the actual UCB swaps for SWAP or splitting off the target volumes for SWAPDUMP. You indicate that it is time to complete the operation by running a CONFIRM job (an FDRPAS job with the command MONITOR TYPE=CONFIRMSWAP or MONITOR TYPE=CONFIRMSPLIT). For SWAPDUMP to duplicate an LPAR or create a new LPAR before running the CONFIRM job, you should quiesce the activity on your LPARs as much as possible so that no unnecessary jobs or online systems are running. In some cases, quiescing may also be desirable for SWAP.

SYNCHRONIZING THE JES AND COUPLING VOLUMES

In FDRPAS V5.4/80 and above, it is no longer necessary to put JES and coupling volumes into separate jobs. If any JES spool or checkpoint volumes, or volumes containing sysplex coupling data sets, are included in the operation, FDRPAS automatically recognizes that it is desirable not to copy them until after the other volumes. The FDRPAS SWAP or SWAPDUMP job issues an FDR189 message at the beginning of the operation (when `CONFIRMSWAP` or `CONFIRMSPLIT` is specified and `MAXACTIVESWAPS=` is specified) to indicate that copying the JES and coupling volumes is deferred. This message appears only on the ISPF terminal, not on the console, and only on the LPARs where it applies.

If it is not important to complete the operation at a particular time, do not specify `CONFMESS=YES` on the CONFIRM job. Run the CONFIRM job when the other volumes have been synchronized. The CONFIRM job internally replies "GO" to message FDR189. The JES and coupling volumes are then copied, and the CONFIRM job tells the SWAP or SWAPDUMP jobs to complete the operation as soon as all of the volumes are synchronized.

If it is important to complete the operation at a particular time, **do** specify `CONFMESS=YES` on the CONFIRM job. Estimate how long it will take to copy the JES and coupling volumes, double the estimate, and run the CONFIRM job that long before the time you want to complete the operation. (When estimating, take into account that the copies of these volumes may not all run concurrently; volumes that have the same MASID number are serialized.) You may want to plan ahead so that all of the other volumes have been synchronized by the time you run the CONFIRM job. The CONFIRM job internally replies "GO" to message FDR189. The JES and coupling volumes are then copied. When all of the volumes are synchronized, the CONFIRM job issues WTOR message FDRW07 on the console. Reply YES (on each LPAR running a CONFIRM job) when it is time to complete the operation.

GENERATING THE CONFIRM JOB

Generate the job stream for CONFIRM using GENSWAP with the same MOUNT statements as for the SWAP or SWAPDUMP. The `MAXGEN` value for CONFIRM must be larger than the total number of volumes being processed because all volumes must be confirmed by the same job. The reason is that each CONFIRM job must be able to check that all of the volumes are ready to confirm before confirming any of them. If SWAP or SWAPDUMP jobs are being run on multiple LPARs, then a CONFIRM job must be submitted to each of those LPARs and each CONFIRM job must include all of the volumes. Specify `SUBMITDELAY=0` since there is only one job (per LPAR). The special type of routing statement that rotates the submitted jobs among multiple LPARs cannot be used since that format only applies when different generated jobs process different volumes. Instead, a `//*SWAPNEXT` statement is used to tell GENSWAP to generate all of the same MOUNT commands for each CONFIRM job. The job names for the CONFIRM jobs do not contain ampersands since there is only one job per LPAR, but if the CONFIRM jobs are being submitted to multiple LPARs then the job names should contain a character to identify the LPAR.

This example of the GENSWAP for CONFIRM corresponds to the GENSWAP for SWAPDUMP earlier in this section. As usual, an override of the `//PAS.INRDR DD` statement directs the output to SYSOUT for viewing instead of submitting the output directly to the internal reader for execution. Be sure to review the generated jobstream well in advance of the time to run the actual CONFIRM job. When it is time to run the actual CONFIRM job, remove the `//PAS.INTRDR DD` statement to allow the generated jobs to be submitted to the LPARs for execution. This example can be found in the JCL library install with FDRPAS with member name PA30508B.

- Specify `TYPE=CONFIRMSWAP` instead of `TYPE=CONFIRMSPLIT` if the operation is a SWAP.
- `LARGESWAP=` is required in the CONFIRM job when the main SWAP or SWAPDUMP jobs are running on multiple LPARs. `LARGESWAP=` tells the CONFIRM job that it should only CONFIRM the volumes for which the main SWAP or SWAPDUMP job is on the same LPAR.
- Remove `CONFMESS=YES` if it is not important to complete the operation at a particular time.

```
//GENSWAP  EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.INTRDR DD SYSOUT=*
//PAS.SYSIN  DD *
  GENSWAP  TYPE=FULL,MAXGEN=16000,VOLSORT=YES,
           SUBMITDELAY=0,MAXCARDS=16000
  MOUNT    VOL=*,SWAPUNIT=5***,CUID=65950
…
  MOUNT    VOL=*,SWAPUNIT=A***,CUID=22398
/*
//PAS.PASJOB DD DATA,DLM=SS
//*SWAPJOB
//PASCA    JOB (ACCT),'STGADM',MSGCLASS=X,MSGLEVEL=(1,1),CLASS=M
/*ROUTE XEQ JESCPUA
//CONFIRM  EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN  DD *
  MONITOR  TYPE=CONFIRMSPLIT,MAXCARDS=16000,LARGESWAP=16000,
           CONFMESS=YES
  MOUNT    VOL=&&&&&&,SWAPUNIT=&&&&
/*
//*SWAPNEXT
…  (replicate CONFIRM for all LPARs running main SWAPs/SWAPDUMPs)
//*SWAPNEXT
//PASCF    JOB (ACCT),'STGADM',MSGCLASS=X,MSGLEVEL=(1,1),CLASS=M
/*ROUTE XEQ JESCPUF
//CONFIRM  EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN  DD *
  MONITOR  TYPE=CONFIRMSWAP,MAXCARDS=16000,LARGESWAP=16000,
           CONFMESS=YES
  MOUNT    VOL=&&&&&&,SWAPUNIT=&&&&
SS                  <- End of outer PASJOB template
```

For SWAPDUMP, after the SWAPDUMP jobs split off the target volumes and terminate, the target volumes are up-to-date duplicates of your online volumes. The target DASD subsystem can now be used to IPL the duplicate LPAR on another processor or at a remote location, assuming all the necessary volumes to run the LPAR have been duplicated.

(*Only for SWAPDUMP*.) If the `LARGERSIZE=OK` operand was used to do the SWAPDUMPs to target volumes that are larger than the source volumes, then the VTOC index is disabled on the target volumes, and the DOS flag is set in the VTOC to indicate that the free space information is invalid. An FDRPAS SWAP task rebuilds the VTOC index, but SWAPDUMP is unable to do so since the target volume remains offline. The first time an allocation is done to such a target volume, z/OS rebuilds the free space information in the VTOC. The VTOC index can be rebuilt by using the SWAPBUILDIX command of FDRPAS as shown in this manual in Section 310.13 "SWAPBUILDIX and EXPANDVTOC Statement".

305.9 GENSWAP CONSIDERATIONS

ISPF DISPLAY NOTES

When you first enter the FDRPAS ISPF panels, press ENTER to display all active SWAP tasks (whenever you press ENTER again, it updates the display and adds any volumes that have recently become active).

Then type the SORT command on the command line. In the sort field display that appears, next to the STATUS field type "1 A" to sort on the status field. This way, volumes that are actively copying data appear first in the display, volumes that are in "CONFIRM" status appear next, and inactive volumes last.

By default, the ISPF display shows a number of lines for each volume, which can limit the number of volumes displayed at a time. Use PF10 (LEFT) or PF11 (RIGHT) to switch to alternate views that display more volumes per screen.

WAITSWAPTERM NOTES

If it is necessary to ensure that the swap of all the volumes have completed before performing another process on these volumes (for example, restarting HyperSwap or AutoSwap), then a `MONITOR TYPE=WAITSWAPTERM` job should be used. See Section 310.9 "MONITOR WAITSWAPTERM Statement" for the command syntax and examples of usage in Section 320.6 "FDRPAS and IBM GDPS/PPRC HyperSwap" and Section 320.8 "FDRPAS and EMC AutoSwap".

DEALING WITH POSSIBLE PROBLEMS

If a volumes has very high activity and the I/O from SWAP or SWAPDUMP is interfering with the user of the volume by applications, then you can:

- Suspend the active SWAP or SWAPDUMP by specifying `SUSPEND` on the ISPF panel. When you are ready to resume the SWAP or SWAPDUMP process, specify `RESUME` on the ISPF panel.
- Slow down the I/O from SWAP or SWAPDUMP by specifying `PACEDELAY` on the ISPF panel.
- Terminate processing of the volume by specifying `ABORT` on the ISPF panel. In this case, you will want to resubmit the SWAP or SWAPDUMP job for this volume at a later time, probably close to the time of running the CONFIRM job. If you aborted this volume or had volumes fail, you can resubmit the GENSWAP that generated the SWAP or SWAPDUMP jobs using the original MOUNT statements by adding an additional operand of `ALLREADYINSWAP=BYPASS` to the GENSWAP statement instructing the GENSWAP process to bypass any volumes that are already in SWAP status. Also, specify `SWAPID=` with a number higher than the highest number generated by the original SWAP or SWAPDUMP jobs.

THIS PAGE INTENTIONALLY LEFT BLANK

310 FDRPAS SWAP JCL, STATEMENTS, EXAMPLES, AND ISPF INTERFACE

310.1 FDRPAS JOB CONTROL REQUIREMENTS

The following JCL statements are required to execute FDRPAS SWAP and MONITOR tasks as batch jobs or started tasks. Please review the examples:

- Section 310.20 "LICENSE Examples",
- Section 310.21 "FDRPAS SWAP Task Examples",
- Section 310.22 "SWAPDUMP Examples",
- Section 310.23 "MONITOR SWAP Examples",
- Section 310.24 "MONITOR CONFIRM Examples", and
- Section 310.25 "HISTORY Examples")

for a better understanding of the JCL requirements of FDRPAS.

RECOMMENDATION: *INNOVATION DATA PROCESSING recommends that you use GENSWAP if you have hundreds or more volumes or an entire DASD box to replace. See Chapter 305 "FDRPAS GENSWAP JCL, Statements, and Examples" for more information.*

EXEC STATEMENT Specifies the program name (PGM=FDRPAS), region requirement (REGION=0M is recommended), and optional PARM= field that may contain the first FDRPAS control statement. For example,

```
//SWAP     EXEC PGM=FDRPAS,REGION=0M
//          PARM='SWAP TYPE=FULL,LARGERSIZE=OK'
```

The parameter (PARM) may also contain a MOUNT control statement, separated from the first by a slash (/), for example,

```
PARM='SWAP TYPE=FULL/MOUNT VOL=PROD01,SWAPUNIT=17F4'
```

NOTE: There must be no space immediately before the slash. Additional control statements, if any, must be contained in the SYSIN data set.

STEPLIB OR JOBLIB DD STATEMENT Specifies the load library where FDRPAS resides. The library must be authorized. FDRPAS must be executed with a JOBLIB or STEPLIB, it should never be put into the system linklist.

SYSPRINT DD STATEMENT Specifies the output message data set; it is required. It is usually a SYSOUT data set but if it is assigned to a data set on tape or DASD, this DD statement must specify DISP=MOD. DCB characteristics are RECFM=FBA and LRECL=121; the block size defaults to 1210 on DASD or tape.

SYSPRIN*x* DD STATEMENT Specifies an output message data set for an internal swap subtask; they are used only when MAXTASKS=*nn* is specified. "*x*" is 1-9, 0, and A-V, in that order, depending on the value of MAXTASKS=*nn*. SYSPRIN*x* DD statements are optional. If needed, SYSPRIN*x* DD statements are dynamically allocated as SYSOUT=*, so you need to specify SYSPRIN*x* DD statements only if you need to direct those messages elsewhere. It is usually a SYSOUT data set but if it is assigned to a data set on tape or DASD, this DD statement must specify DISP=MOD. DCB characteristics are RECFM=FBA and LRECL=121; the block size defaults to 1210 on DASD or tape.

SYSPRT*xx* DD STATEMENT Specifies an output message data set for an internal monitor subtask. SYSPRT*xx* DD statements are used only in an FDRPAS step with a MONITOR TYPE=SWAP control statement. SYSPRT*xx* DD statements are optional. If needed, SYSPRT*xx* DD statements are dynamically allocated as SYSOUT=*, so you need to specify them only if you need to direct those messages elsewhere. It is usually a SYSOUT data set but if it is assigned to a data set on tape or DASD, this DD statement must specify DISP=MOD. DCB characteristics are RECFM=FBA and LRECL=121; the block size defaults to 1210 on DASD or tape.

FDREMAIL DD STATEMENT Specifies input control statements for the FDR e-mail facility. If present, e-mail messages can be sent for unsuccessful or successful FDRPAS operations. See Section 320.4 "FDRPAS E-mail Notification Facility" for requirements and details.

FDRSUMM DD STATEMENT

Specifies the optional output message data set for an FDRPAS summary report. It is usually a SYSOUT data set but if it is assigned to a data set on tape or DASD, this DD statement must specify `DISP=MOD`. DCB characteristics are `RECFM=FBA` and `LRECL=121`; the block size defaults to 1210 on DASD or tape. The FDRSUMM DD statement contains a one-line summary for each volume processed. It is ignored for MONITOR tasks.

SYSUDUMP DD STATEMENT

Specifies the ABEND dump data set. Usually specifies a SYSOUT data set. Although the SYSUDUMP DD statement is not required, we strongly urge you to always include this DD statement, so that we can help you diagnose error conditions. If you have a debugging aid product on your system that would prevent the desired dump, please add the appropriate one of these statements to the JCL so that a fully-formatted dump is produced.

```
//ABNLIGNR DD DUMMY  Turn off Abend-Aid
//CAOESTOP DD DUMMY  Turn off CA OPT II & CA SYMDUMP
//DMBENAN  DD DUMMY  Turn off DumpMaster
//ESPYIBM  DD DUMMY  Turn off Eye-Spy
//IDIOFF   DD DUMMY  Turn off IBM Fault Analyzer
//PSPOFF   DD DUMMY  Turn off Softworks Performance Essential
```

SYSIN DD STATEMENT

Specifies the control statement data set. Usually an input stream or DD * data set. It may be DUMMY if all necessary control statements are included in the `PARM=` on the EXEC statement.

PASJOB DD STATEMENT

Specifies the input control statement for the FDRPAS job submission facility. If present, it is used to submit FDRPAS MONITOR tasks to the other LPARs where MONITOR tasks are not already running. The input is a set of JCL statements necessary to route and execute a job to the desired LPAR(s). Each "set" is proceeded by a `//*CPUID=` statement that designates the CPUID where that set of statements are to be sent. JCL statements are not submitted to the CPU where the main job is running or where a MONITOR task is already running. It is your responsibility to code the necessary JCL statements (`/*ROUTE` and `/*JOBPARM SYSAFF=`) and parameters (`CLASS=`) needed to route the job to the correct CPUs.

NOTE: If a CPU is not active, FDRPAS submits the MONITOR task but it does not execute until that CPU is started. This is not a problem for FDRPAS as it does not need a monitor on a system that is completely down. However, this MONITOR task executes when the CPU comes up again and you may have to STOP (P) it.

The CPUID value is the 10-character CPU serial number of a system image. When you run a SIMSWAP job, the CPUID values display on the FDR233 message as shown in this example:

```
FDR303  CARD IMAGE --   SIMSWAP  TYPE=FULL
FDR303  CARD IMAGE --   MOUNT    VOL=HI17C2,SWAPUNIT=17C1
FDR233  CPU WITH (SERIAL# 0212342818) IS ATTACHED TO VOL=HI17C2            - HTC 2107900 TO HTC 2107900
FDR233  CPU WITH (SERIAL# 0112342818) IS ATTACHED TO VOL=HI17C2            - HTC 2107900 TO HTC 2107900
FDR233  CPU WITH (SERIAL# 0512342818) IS ATTACHED TO VOL=HI17C2            - HTC 2107900 TO HTC 2107900
FDR233  CPU WITH (SERIAL# 0912342818) IS ATTACHED TO VOL=HI17C2            - HTC 2107900 TO HTC 2107900
FDRW66  SWAP OF VOL=HI17C2 TO UNIT=17C1 NEEDS TO BE STARTED ON   4 SYSTEMS
```

You can also get the CPUID value is the on a specific z/OS system by executing this console command from a console that is attached to the system:

```
D M=CPU
```

You receive a response similar to:

D M=CPU

```
IEE174I 15.34.53 DISPLAY M 899
PROCESSOR STATUS
ID  CPU                SERIAL
00  +                   0212342818
01  +                   0212342818
```

The following is an example of using PASJOB to route MONITOR tasks to two different systems that uses a `/*ROUTE` statement to route the MONITOR tasks. You may have different requirements to route jobs such as `CLASS=`, `SYSAFF=`, or a `ROUTE` statement to get the job to the proper LPAR in the SYSPLEX.

```
//PASJOB   DD DATA,DLM=$$
//*CPUID=0912342818
//PASMONA  JOB …
/*ROUTE    XEQ  JESCPUA
//MONITOR  EXEC PASPROC
//PAS.SYSIN DD *
  MONITOR  TYPE=SWAP,DYNMON=YES
/*
//*CPUID=0212342818
//PASMONB  JOB …
/*ROUTE    XEQ  JESCPUB
//MONITOR  EXEC PASPROC
//PAS.SYSIN DD *
  MONITOR  TYPE=SWAP,DYNMON=YES
/*
$$
```

USING PASPROC TO EXECUTE FDRPAS

As part of the installation of FDRPAS, you were required to edit a cataloged procedure (PROC) for FDRPAS and store it in a system procedure library (see "PASPROC" in Section 380.16). This was required so that the FDRPAS MONITOR task can start additional MONITOR tasks as system started tasks. However, you can also use this PROC to execute FDRPAS in batch jobs or to start FDRPAS from a system console with a `START (S)` command. The default name of this PROC is PASPROC but if you changed it during installation, substitute your name in the examples in this manual.

In a batch job, use JCL such as:

```
//SWAP     EXEC PASPROC
//PAS.SYSIN DD *
  SWAP     TYPE=FULL
  MOUNT    VOL=TS0123,SWAPUNIT=1234
/*
```

From the console, use syntax such as:

```
S PASPROC.VTSO123,PARM='SWAP TYPE=FULL/MOUNT VOL=TSO123,SWAPUNIT=1234'
```

If your systems are part of a sysplex (basic or parallel), you can use the console ROUTE command to start MONITOR tasks on all systems in the sysplex. The ROUTE command can considerably reduce the amount of typing required. For example:

```
ROUTE T=0,*OTHER,S PASPROC.MON2,PARM='MONITOR TYPE=SWAP/MOUNT SU=2*'
```

starts the MONITOR task on all other systems in the sysplex (`*OTHER` assumes that you start the SWAP task on this system, use `*ALL` to start the MONITOR task on all systems).

310.2 SWAP, SWAPDUMP, SIMSWAP, SIMSWAPMON STATEMENTS

SWAP, SWAPDUMP, SIMSWAP, AND SIMSWAPMON STATEMENT SYNTAX

The syntax of the SWAP, SWAPDUMP, SIMSWAP, and SIMSWAPMON statements:

```
SWAP
SWAPDUMP
SIMSWAP
SIMSWAPMON
```

```
TYPE=FULL

,ALLOWPAV=NO|
          YES

,BUFNO=32|
       nn

,CANCELPROT=NO|
            YES

,CHECKSOURCE=NO|
             YES

,CHECKTARGET=NO|
             YES

,CONFIRMSPLIT=NO|
              YES

,CONFIRMSWAP=NO|
             YES

,CONFMESS=NO|
          YES

,EMSG=(OK,nnn)

,LABEL=NOSWAP|
       SWAP

,LARGERSIZE=NO|
            OK

,LOGMESS=NO|
         YES

,MAXCARDS=250|
          nnnn

,MAXACTIVESWAPS=NO|
                YES

,MAXTASKS=1|
          nn

,NOTIFYERR=userid

,PACEDELAY=0|
           nnnnn

,PACING=DYNAMIC|
        STATIC

,PRINT=ALL

,PRTDEFAULTS

,NONRESPONDING=FAIL|
               MSG|
               RETRY

,SRVCLASS=classname|
          HIGHEST>|
          NULL

,SWAPDELAY=nnn

,SWAPIOERR=NORETRY|
           RETRY

,VOLSORT=NO|
         YES

,WTOR=NO|
      YES
```

ADDITIONAL LARGE SCALE SWAP OPERANDS

Additional operands for Large Scale swaps with CONFIRM:

```
,COPYVOLID=NO|
           YES

,LARGESWAP=nnnnn

,SWAPID=nnn
```

SWAP STATEMENT

This statement initiates an FDRPAS SWAP task to move a volume to a new DASD device. It must be the first statement in the input; only one SWAP statement is allowed per execution. SWAP must be followed by one or more MOUNT statements to identify the online volumes to be swapped and may optionally be followed by one or more EXCLUDE statements to exclude certain systems that do not have a source volume online.

If multiple MOUNT statements are specified, by default FDRPAS processes them serially, one at a time. However, you can request that this SWAP task process multiple volumes concurrently, up to 64 at a time, by specifying the `MAXTASKS=nn` operand. To swap more than 64 volumes concurrently, you must start multiple SWAP tasks.

A RACROUTE call is issued to verify that the user has READ authority to resource "FDRPAS.SWAP" in the FACILITY class, if that resource is protected. If "FDRPAS.SWAP" is not protected, the operation continues.

If you have security checking enabled (see "ALLCALL" in Section 380.14), SWAP checks that the user has ALTER authority to the source volume under the DASDVOL security class. If the user does not have DASDVOL authority, the operation fails. If the volume is not protected by DASDVOL, FDRPAS checks that the user has ALTER authority in the DATASET class to every data set on the volume; if any data set is not authorized the operation fails.

NOTE: Target devices for SWAP tasks must be offline before running the SWAP task. If the target device is not offline on an LPAR that a MONITOR task is running, special checking is done by the MONITOR task to ensure that this device is the same target device as specified by the main FDRPAS process and that the device is inactive on the LPAR this MONITOR task is running on. If so, then this volume is varied offline on by this MONITOR task.; otherwise, it is left online.

SWAPDUMP STATEMENT

SWAPDUMP is similar to SWAP, except that the volume is not swapped to the target device. It is used to create a point-in-time copy of the source volume on the target device that can then be backed up with FDRINSTANT. All rules and considerations of SWAP also apply to SWAPDUMP. Contact INNOVATION DATA PROCESSING if processing a large number of volumes to go over GENSWAP.

Start the SWAPDUMP operation far enough ahead of the time you plan to do the backup so that FDRPAS can copy the data and synchronize the volumes. If you specify `CONFIRMSPLIT=YES`, FDRPAS then continues to keep the volumes synchronized until you "confirm" the operation and create the frozen point-in-time copy, at that time you can submit the FDRINSTANT backup jobs.

`MAXTASKS=nn` is especially useful for SWAPDUMP when there is the need to create point-in-time backups of a large number of DASD volumes all at the same point-in-time. `MAXTASKS=nn` allows the SWAPDUMP of up to 64 DASD volumes in a single swap job or started task, so that the number of jobs or tasks required to synchronize all of those volumes is small.

A RACROUTE call is issued to verify that the user has READ authority to resource FDRPAS.SWAPDUMP in the FACILITY class, if that resource is protected. If FDRPAS.SWAPDUMP is not protected, the operation is allowed.

NOTE: If you have security checking enabled (see "ALLCALL" in Section 380.14), SWAPDUMP checks that the user has at least READ authority to the source volume under the DASDVOL security class. If the user does not have DASDVOL authority, the operation fails. If the volume is not protected by DASDVOL, FDRPAS checks that the user has READ authority in the DATASET class to every data set on the volume; if any data set is not authorized, the operation fails.Target devices for main SWAP tasks must be offline before running the SWAP task. If the target device is not offline on an LPAR that a MONITOR task is running, special checking is done by the MONITOR task to ensure that this device is the same target device as specified by the main FDRPAS process and that the device is inactive on the LPAR this MONITOR task is running on. If so, then this volume is varied offline on by this MONITOR task, otherwise, it is left online to that MONITOR task.

CAUTION: For SWAPDUMP Only!
INNOVATION recommends that users NOT include JES SPOOL, CHECKPOINT, and Sysplex Coupling Facility volumes in SWAPDUMP and MONITOR jobs along with other volumes; and recommend that JES SPOOL, CHECKPOINT, and Sysplex Coupling Facility volumes be put in individual jobs (each volume by itself) with no other volumes, using a separate MONITOR job for each JES SPOOL, CHECKPOINT, and Sysplex Coupling Facility volume, and running these jobs after all the other SWAPDUMP jobs are complete.

SIMSWAP STATEMENT

SIMSWAP performs a simulation of a SWAP operation. The SIMSWAP statement accepts all of the operands and statements of a SWAP operation but it does not actually perform a swap. The SIMSWAP statement does not require MONITOR tasks on other systems; if MONITOR tasks are running they do not participate.

SIMSWAP has these uses:

- It validates all of the operands that appear on the SWAP statement.
- It validates the online volumes and offline target devices specified on the MOUNT statements on the system where the SIMSWAP is executed. This identifies errors such as the target device does not exist or is not offline, or is the wrong type or size.
- For DASD volumes in subsystems where FDRPAS can determine the systems with access to the source volume, it displays all of the identified systems, including their CPU IDs, in FDR233 messages. You can verify that all of the systems you expect have access, and that there are no unexpected systems with access.

RECOMMENDATION: *Run SIMSWAP once to acquire all the CPU IDs needed to run the swap. Following that, we recommend using SIMSWAPMON to perform a more comprehensive simulation of the SWAP and MONITOR tasks.*

SIMSWAPMON STATEMENT

SIMSWAPMON performs a simulation of a SWAP operation and provides additional processing above that performed by SIMSWAP by communicating with the MONITOR tasks. The SIMSWAPMON statement accepts all of the operands and statements of a SWAP operation so they can be used for the actual SWAP operation after running the SIMSWAPMON. The SIMSWAPMON does not perform the SWAP. SIMSWAPMON requires and verifies that MONITOR tasks are running on all the other systems.

SIMSWAPMON has these uses:

- It validates all of the operands that appear on the SWAP statement.
- It validates the source volumes and target devices specified on the MOUNT statements on the system where SIMSWAPMON is executed as well as all the systems where MONITOR tasks are running. This identifies errors such as the target device does not exist, or is the wrong type or size.
- It displays all the systems that respond and that join in the swap operation. It also displays all the systems that did not join (ERROR) or did not need to join (OK). Non-responding systems where MONITOR tasks are not running are also displayed.
- It can check the integrity of the VTOC and VVDS on the source volume (`CHECKSOURCE=YES`), and whether the target device is empty of data sets (`CHECKTARGET=YES`).
- It runs up to 32 tasks at a time.
- It is recommended to always run SIMSWAPMON before running any dynamic processing to allow SIMSWAPMON to populate the MONITOR tasks (if needed).

The time needed to validate the devices depends on how the devices are specified or discovered in the MONITOR tasks. This is the time that it takes for the SWAP task to discover the status of a non-responding device. For example,

- It takes approximately 5-10 seconds for every 32 devices when the MONITOR tasks have MOUNT statements with the appropriate SWAPUNITs.
- It takes approximately 30-40 seconds for every 32 devices when the MONITOR task dynamically adds the SWAPUNIT.
- It takes approximately 120 seconds for every 32 devices when one or more MONITOR tasks are not running or cannot dynamically add the SWAPUNIT.

NOTE: Certain situations cannot be identified until the actual SWAP is attempted. These include having a duplicate source volume specified or having duplicate SWAPUNITs specified.

SWAP, SWAPDUMP, SIMSWAP, AND SIMSWAPMON STATEMENT OPERANDS

TYPE=FULL

Must be specified on the SWAP, SWAPDUMP, SIMSWAP, or SIMSWAPMON statement.

#SYSTEMS=

NSYSTEMS=

NOTE: The #SYSTEMS= and the equivalent NSYSTEMS= operands are no longer documented and INNOVATION DATA PROCESSING recommends that it NOT be used unless you are swapping a really old control unit. Contact INNOVATION DATA PROCESSING if you receive an FDR234 REASON=R message indicating that you need to specify #SYSTEMS.

ALLOWPAV=

The ALLOWPAV= operand controls the status of the Parallel Access Volume (PAV) aliases during the SWAP operation. This operand specifies whether the Parallel Access Volume (PAV) aliases for the source volume are disabled at the start of the SWAP operation or at the time when the actual SWAP of the volume occurs. (The PAV aliases for the target volume are always disabled at the start of the SWAP operation.)

NO – Indicates that the Parallel Access Volume (PAV) aliases for the source volume are disabled at the start of the SWAP operation and re-enabled after the SWAP operation completes.

YES – Indicates that the Parallel Access Volume (PAV) aliases for the source volume stay enabled during the initial copy phase and are disabled during the time the actual SWAP of the volume occurs. This reduces the amount of time that PAV aliases are not available.

Default: NO.

NOTE: ALLOWPAV=YES needs to be specified on all of the SWAP and MONITOR tasks that you wish to leave PAV active.

BUFNO=

***nn* –** Specifies the number of I/O buffers that FDRPAS uses while copying data from the source volume to the target device, from 2 to 32. It should be an even number and is rounded up if odd. FDRPAS divides this buffer set in half, in order to overlap input and output I/Os. With BUFNO=32, FDRPAS reads and writes up to 15 tracks per I/O; with lesser values, it does half of the BUFNO= value in tracks per I/O. A smaller BUFNO= value reduces the elapsed time of each I/O, and thus reduces the impact of each I/O on your I/O system, but it increases the number of I/Os required to copy the data, and thus increases the total time to do the swap.

INNOVATION DATA PROCESSING does not recommend specifying BUFNO= unless the length of the FDRPAS I/Os are causing problems. For example, certain hardware channel extenders may experience errors on extremely long I/O chains, causing the swap to fail. Reducing BUFNO= may allow the swap to work. If many FDRPAS swaps are running concurrently, the length of each individual FDRPAS I/O may impact overall system performance; reducing BUFNO= may decrease this impact (see also PACEDELAY=).

Default: 32.

CANCELPROT=

NO – A `CANCEL (C)` command causes immediate termination of the SWAP task.

YES – FDRPAS protects this SWAP task against accidental cancellation. If a `CANCEL (C)` command is issued, it is treated like a `STOP (P)` command. Message FDR267 is issued and the task lets active swaps complete before terminating. You can issue a second `CANCEL (C)` command to force immediate termination.

Default: NO.

CHECKSOURCE=

FDRPAS checks the integrity of the VTOC and VVDS. It is recommended that a `CHECKSOURCE=YES` be run at least once before running the actual swap to check the source volumes for errors

NO – The source volume is not checked for structural errors in the VTOC and VVDS.

YES – During the SIMSWAP or SIMSWAPMON process, FDRPAS checks the source volume for structural errors in the VTOC and VVDS. This operand is ignored on SWAP and SWAPDUMP operations.

NOTE: Do not specify `CHECKSOURCE=YES` for z/VM, Linux for System z, and CMS volumes.

Default: NO. `CHECKSOURCE=YES` is recommended for SIMSWAPMON jobs.

CHECKTARGET=

NO – The target volume is not checked. The target volume is overlaid regardless of its current contents.

YES – FDRPAS checks the target device before beginning a SWAP or SWAPDUMP operation to ensure that the target is empty. The operation is terminated if the target contains any data sets other than a VTOC, VTOC index (SYS1.VTOCIX.*volser*), or VVDS (SYS1.VVDS.V*volser*). Also, volumes initialized by z/VM, with a dummy VTOC in cylinder 0 track 0, are bypassed. DASD volumes that do not contain a valid volume label (such as those that have never been used since they were delivered or defined) are also accepted; this includes volumes that were the source volumes of successful FDRPAS swaps.

Default: NO. `CHECKTARGET=YES` is recommended for SIMSWAPMON jobs.

NOTE: FDRPAS cannot verify that a target device is offline to every system, so `CHECKTARGET=YES` can be used to ensure that you are not overlaying a volume with valid data on it. However, you must either delete all existing data sets from the target volume or initialize it with ICKDSF so that it appears to be empty. If you are using FDRPAS for I/O load balancing, do not specify `CHECKTARGET=YES` since the target device probably contains data sets from its previous use.

CONFIRMSPLIT=

CONFIRMSWAP=

CONFIRMSPLIT= is used only with the SWAPDUMP statement, and CONFIRMSWAP= is used only with the SWAP statement, but they function identically. CO= is a valid abbreviation for either operand. **CONFIRMSWAP=YES should be used only when you have a need to complete the swap of a number of volumes at the same time; if it is not important when the swap of each individual volume completes, use the default of CONFIRMSWAP=NO.**

NO – FDRPAS completes the operation as soon as the source and target devices are in synchronization, without waiting for any confirmation. CONFIRMSWAP=NO is recommended for a SWAP operation unless you have a need to swap a set of volumes at the same time.

YES – FDRPAS does not complete the swap or dump of this volume until you confirm that you are ready to do so. The operation continues through Phases 1 through 4 (see "SWAP Phases" in Section 300.3) and then waits, copying updated tracks as required, until you confirm that the operation is to be completed, by using the FDRPAS ISPF panels (see Section 310.31 "FDRPAS ISPF Interface") or a MONITOR TYPE=CONFIRMSWAP or CONFIRMSPLIT job (see Section 310.7 "MONITOR CONFIRM Statement"). With SWAPDUMP, CONFIRMSPLIT=YES can be used to continue updating the duplicate volume until you are ready to create the point-in-time backup.

Default: NO.

NOTE: CONFIRMSPLIT=YES and CONFIRMSWAP=YES do not result in any console message or WTOR when the volumes are ready for the swap or split to complete. You can only tell when the volumes are in this state by using the FDRPAS ISPF panels, and you can confirm the operation only by using the ISPF panels or the MONITOR TYPE=CONFIRMSWAP or CONFIRMSPLIT statement.

CONFMESS=

CM=

YES – Before beginning the swap, FDRPAS requests confirmation via FDRW01 message that must be replied to before the swap can start. The FDRW01 can be issued as a WTOR to the system operator (see "WTOR=" in Section 310.2 operand) or you can reply to this message from the FDRPAS ISPF panels.

NO – Suppresses the WTOR and begins the swap immediately.

Default: NO.

CPYVOLID=

Specifies whether the volume label of the target DASD volume should be copied from the source volume for SWAPDUMP process.

NO – The volume label of the target volume is left as "FDR1' so it cannot be brought online.

YES – The volume label of the target volume is set to "VOL1" so that it can be brought online.

Default: NO.

EMSG=

When the FDR e-mail notification facility has been invoked by including an FDREMAIL DD statement in the FDRPAS execution JCL, EMSG= is used to modify the conditions when an e-mail is sent. The parameter is a list of 3-digit FDR message numbers in parenthesis, and may also include the keyword "OK". The parenthesis can be omitted if only one parameter is specified. Examples:

```
EMSG=123
EMSG=(123,456)
EMSG=(OK,123,456,789)
EMSG=OK
```

OK – E-mail notifications are sent for successful SWAP and SWAPDUMP operations as well as failures. Note that this generates a separate e-mail for every DASD volume processed by FDRPAS, which could result in many messages.

***nnn* –** E-mailed messages include message FDR*nnn* (if it is issued). FDRPAS has a fixed set of such messages (FDR302, FDR316, FDR319, FDR234, and FDR997), EMSG=*nnn* adds to the list. Note that EMSG=999 generates a single message for the whole FDRPAS step if the FDR999 (successful) message is issued; separate messages for failed swaps are still sent.

Default: E-mail notifications are sent only for SWAP and SWAPDUMP failures.

LABEL=

NOSWAP – On a SWAP process, the volume serial of the source volume is unchanged, and the volume label identifier of the source volume is set to a special FDRPAS designation so that the volume cannot be brought online. On a SWAPDUMP process, the volume serial of the target volume is copied from the source volume, and the volume label identifier of the target volume is set to a special FDRPAS designation so that the volume cannot be brought online.

SWAP – On a SWAP process, the volume labels of the target and source volumes are interchanged. On a SWAPDUMP process, the volume label of the target volume is retained.

- For a successful SWAP process:
  - If the target volume had a valid volume label before the SWAP process, that volume label is copied to the source volume, A valid volume label is a label that contains "VOL1" or "RSV1" in the volume label identifier field. "RSV1" indicates that the volume was initialized by the ICKDSF INIT command with the RESERVED operand.
  - The target volume label is copied from the source volume.
- For a successful SWAPDUMP process:
  - The source volume label is unchanged.
  - If the target volume had a valid volume label before the SWAPDUMP process, that volume label is retained,

RECOMMENDATION: *For HyperSwap, we recommend initializing the target volumes as RESERVED so the volumes can be offline in the GDPS system prior to the SWAP.*

NOTE: IBM APAR **PM16856** provides the new function to the ICKDSF INIT command to mark a volume as a RESERVED volume. A volume that is reserved has a volume serial number but cannot come online to any system.

Default: NOSWAP.

LARGERSIZE=

NO – The output device must have the same number of data cylinders as the source volume. The output device does not have to have the same number of alternate cylinders as the source volume. If the number of alternates is different, FDRPAS corrects the alternate count after the swap.

OK – The target device may have more data cylinders than the source volume. `LARGERSIZE=OK` allows you to swap from one model of a DASD volume to a larger model. For example, you can swap from a 3390-3 (3339 cylinders) to a 3390-9 (10017 cylinders). The volume size is updated in the VTOC and VTOCIX (if active), as well as all in-storage tables, when the devices are swapped. However, there is a small chance that the VTOCIX index may be disabled during the swap; if so; you can use the FDRPAS SWAPBUILDIX utility function (see Section 310.13 "SWAPBUILDIX and EXPANDVTOC Statement") to rebuild it even while the volume is in use on multiple systems. Note that if the target device is larger than the source, you receive an FDRW66 message with the "OVERRIDE OF WARNINGS" text. FDRPAS invokes ICKDSF to update the volume size.

NOTE: Since the target device remains offline when doing a SWAPDUMP, the VTOCIX (if present) is disabled and the "DOS" flag is set in the F4 DSCB if the target device is a larger DASD volume. If the target volume is subsequently varied online to some LPAR, then the first new data set allocation causes the free space to be update, but the VTOCIX remains disabled. To refresh the VTOCIX you must use FDRPAS "SWAPBUILDIX" or ICKDSF "BUILDIX".

Default: NO.

LARGESWAP=

***nnnnn* –** Specifies a value slightly larger than the number of volumes being swapped, from 2000 to 16000. The purpose of this operand is to reduce the below-the-line CSA requirements. This operand should only be used on a SWAP or SWAPDUMP of more than 2000 volumes. For example, for 5000 volumes, specify `LARGESWAP=6000`. When `LARGESWAP=` specifies a value larger than the number of volumes being swapped, the requirement for below-the-line CSA is no more than 400 KB.

NOTE: This option should only be used with Large Scale Swaps specifying `CONFIRMSPLIT=` or `CONFIRMSWAP=`. See Section 305.8 "FDRPAS for Large Scale Synchronized Migration" for more information and usage examples.

LOGMESS=

NO – No SYSLOG/console messages are written.

YES – Messages are written to SYSLOG (and usually to an operator console) documenting that the swap is occurring, and has completed.

Default: YES for the SWAP statement and NO for the SWAPDUMP statement.

MAXCARDS=

Specifies the maximum number of MOUNT statements that can be present in this FDRPAS step, from 1 to 9999.

NOTE: FDRPAS acquires a table with a size of 160*MAXCARDS bytes in below-the-line storage, so very large values may cause GETMAIN failures. However, values up to 3000 should not be a problem in FDRPAS.

Default: 250.

MAXACTIVESWAPS=

NO – The total number of active SWAP or SWAPDUMP tasks, in multiple FDRPAS jobs, is not limited or controlled. If you submit multiple FDRPAS jobs, each of them may be actively copying data for multiple DASD volumes (up to the MAXTASKS=*nn* limit in each job). Many such jobs may overload DASD channels and system common storage. In this case, INNOVATION DATA PROCESSING recommends that you submit a few such jobs at one time, and submit new jobs as each old job finishes.

YES – FDRPAS limits the number of SWAP or SWAPDUMP tasks in multiple FDRPAS jobs that can be in the initial "Phase 3 pass 1" copy phase, where the tracks identified by FDRPAS as "in-use" are copied. The limit is equal to the value of MAXTASKS=*nn* (see "MAXTASKS=" in Section 310.2) that has a maximum of 64. If you submit multiple FDRPAS jobs, FDRPAS limits to "*nn*" the number of volumes that can be in "pass 1" across all of those jobs. This allows you to submit many FDRPAS jobs while limiting the number of volumes that are actively copying data. Beyond "pass 1", FDRPAS periodically copies updated tracks that are a much smaller load on the system.

RECOMMENDATION: When using MAXACTIVESWAPS=YES, the volumes should be grouped by size in the FDRPAS jobs and the jobs with the largest volumes started first. For example, when swapping a group of volumes that contains a mixture of 3390-54, 3390-27, and 3390-9 volumes, arrange the FDRPAS jobs to have a multiple of the MAXTASKS= value and arrange these groups of volumes by largest to smallest in size. The GENSWAP process does this automatically.

MAXACTIVESWAPS=YES is especially useful when CONFIRMSWAP=YES or CONFIRMSPLIT=YES is specified, since this usually implies many volumes are in "confirm" state before the operation is finally confirmed and terminated. This allows you to submit the FDRPAS jobs for all those volumes at once knowing that no more than 64 are ever in "pass 1".

NOTE: When coding the MAXACTIVESWAPS= operand in the GENSWAP process, do not specify the MAXACTIVESWAPS= operand in the FDRPAS main process model as it may cause undesired consequences with the GENSWAP process.

Default: NO.

MAXTASKS=

nn **–** Specifies the maximum number of volumes that can be processed concurrently by this SWAP task; from 1 to 64. You must follow the SWAP statement with multiple MOUNT statements, specifying the volumes to be processed. If the number of MOUNT statements exceeds MAXTASKS=*nn*, FDRPAS starts the indicated number of swaps; as each one finishes another one is started, until all MOUNT statements have been processed. The FDRPAS ISPF panels can be used to terminate active swaps individually.

In order to separate the messages from these swaps, the messages are written to SYSPRIN*x* DD statements ("*x*" is 1-9, 0, and A-V in that order, depending on MAXTASKS=*nn*). If you have not provided these DD statements in the SWAP task JCL, they are dynamically allocated as SYSOUT=*. At the termination of each swap subtasks, most of its messages are also written to SYSPRINT so that all FDRPAS messages are in one place.

Default: SIMSWAP always uses a MAXTASKS value of 1. SIMSWAPMON always uses a MAXTASKS value of 32. For SWAP and SWAPDUMP, the default is that only one volume is processed at a time. If multiple MOUNT statements are provided, they are processed serially. All messages are written only to SYSPRINT. The MAXACTIVESWAPS= operand can be used to coordinate the total number of active SWAP jobs when multiple concurrent SWAP jobs are running.

RECOMMENDATION: *For SIMSWAPMON, code the MAXTASKS= value that you intend on using for the actual SWAP or SWAPDUMP process to allow the SIMSWAPMON task to validate the value. The SIMWAPMON task does not use the value coded for MAXTASKS=.*

NOTE: If you run multiple concurrent swaps, either through MAXTASKS= or by running multiple swap jobs, you should consider the total impact on your system. For example, if you run many swaps against the same source or target control unit, it may overload the control unit or the channels to that control unit. The impact is very dependent on your hardware configuration, so there are no hard guidelines. Most customers run 20 or more concurrent swaps without any issues.

MIN#SYSTEMS=

MINNSYSTEMS=

NOTE: This operand is no longer documented and INNOVATION DATA PROCESSING recommends that it NOT be used. The EXCLUDE CPUID= statement (see Section 310.4 "SWAP Task EXCLUDE Statement") is the recommended way of handling systems or LPARs that are connected to the source DASD device, but are not z/OS systems or the target devices are not gen'd.

NONRESPONDING=

If a non-responding system is detected during initialization (usually due to a system where no FDRPAS MONITOR is running), then NONRESPONDING= controls how FDRPAS responds.

FAIL – The swap fails immediately, as if "NO" was replied to message FDRW68.

MSG – Message FDRW68 is issued, allowing the user to override the error "YES", terminate the swap "N"), or retry the missing systems "RETRY". See "Phase 1: Initialization" in Section 300.3 for details.

RETRY – Message FDRW68 is issued, allowing the user only to terminate the swap "NO" or retry the missing systems "RETRY". See "Phase 1: Initialization" in Section 300.3 for details.

Default: MSG.

NOTE: If the FDRW68 message is not replied to within 30 minutes, a reply of "NO" is issued automatically.

NOTIFYERR=

***userid* –** Specifies a TSO user ID. Error messages from FDRPAS are sent to that user ID using the SEND command, for immediate notification of problems in FDRPAS execution. If the TSO user is logged on to the same LPAR as the FDRPAS SWAP task, the messages appear as soon as the user presses **ENTER**. On another LPAR, the user must use the LISTBC command to view them, or they are displayed when the user next logs on.

Default: No TSO notification.

PACEDELAY=

***nnnnn* –** Specifies the number of hundredths of a second (1-32767) that FDRPAS waits between WRITE I/Os on the target device, in order to minimize the impact of the background copy operation on other applications (each copy I/O copies up to 15 tracks of source data). In most cases, a value between 1 and 20 should be sufficient. `PACEDELAY` can be dynamically modified for a specific volume from the FDRPAS ISPF panels, see Section 310.31 "FDRPAS ISPF Interface". If `PACING=DYNAMIC` is also specified, the `PACEDELAY=` value is used as the initial pacing delay, and FDRPAS adjusts it dynamically.

Default: 0 (no pacing delay).

PACING=

DYNAMIC – FDRPAS uses an algorithm to gauge the impact of its I/O's on I/O queue length and I/O delay for other tasks on the source volume. Every 15 seconds FDRPAS may adjust the `PACEDELAY=` value in use depending on recent results. If `PACEDELAY=` is also specified, that is used as the initial value; otherwise FDRPAS determines the initial value when the swap starts. If possible, run the swap on the system with the highest level of update activity for the volume when dynamic pacing is used.

STATIC – Static I/O pacing is used for FDRPAS I/O. If `PACEDELAY=` is also specified, FDRPAS inserts the indicated time delay between each write to the target device. The `PACEDELAY` value can be displayed and interactively modified from the FDRPAS ISPF panels.

Default: STATIC. If `PACEDELAY=0` is specified or defaulted, no I/O pacing is done (unless modified from the ISPF panel during the swap).

RECOMMENDATION: INNOVATION DATA PROCESSING recommends that you do not use static or dynamic I/O pacing unless you have experienced unacceptable degradation due to the use of FDRPAS. `PACEDELAY` and dynamic pacing causes the swaps to take longer.

PRINT=

ALL – Requests additional printout from the SWAP task, including a list of all data sets on the source volume, and a detailed list of the tracks copied in each phase.

NOTE: `PRINT=ALL` should not be used when swapping JES spool volumes. Because some messages are printed while I/O is inhibited to the source volume, it may result in an interlock.

PRTDEFAULTS

If specified, the `PRTDEFAULTS` operand lists the defaults for various operands that affect FDRPAS operation. If overriding operands are present before the `PRTDEFAULTS` operand, it displays those overrides. Overriding operands that follow the `PRTDEFAULTS` operand are not displayed.

SRVCLASS=

Allows you to override the service class assigned to the SWAP job by Workload Manager (WLM).

classname **–** The desired service class.

NOTE: `SRVCLASS=SYSTEM` is invalid, because FDRPAS is not eligible to be reset into the SYSTEM service class. However, `SRVCLASS=SYSSTC` is valid.

HIGHEST> – (Note the 8th character is a "greater than" sign.) Set the service class to the same class as the address space with the highest dispatching priority in the system that is X'F8' (248) or less, and that has a single performance period.

NULL – Do not change the service class assigned by WLM.

Default: NULL.

SWAPDELAY=

nnn **–** Specifies the interval in seconds (1-255) that FDRPAS waits between checks to see if all the system images have indicated that they are ready to swap the volume. FDRPAS makes this check up to 30 times; if all expected systems have not joined in the swap by the 30th check, FDRPAS terminates the swap. This may indicate that an FDRPAS MONITOR task was not monitoring the target volume on every system. The value specified should be three or more times the value specified for `SWAPDELAY=` on the MONITOR statement in the MONITOR task.

Default: 15 seconds (the default on the MONITOR statement is five seconds). The default should be adequate unless you have a large number of systems that participate in the swap process; in that case more time may be required for all systems to join.

SWAPID=

nnn **–**

NOTE: This option should only be used with Large Scale Swaps specifying CONFIRMSPLIT= or CONFIRMSWAP=. See Section 305.8 "FDRPAS for Large Scale Synchronized Migration" for more information and usage examples.

SWAPIOERR=

RETRY – An I/O error on the target device while copying tracks causes the swap to stop the copy and retry the I/O error every five seconds until it is successful or until the swap is terminated (by the ABORT command on the FDRPAS ISPF panels or a console `CANCEL (C)` command). `SWAPIOERR=RETRY` allows a swap to recover and continue when the target device has an error that can be corrected. It is intended for use when the target device is connected over a channel extender; if the remote link is interrupted and later recovered, FDRPAS automatically waits for the repair and continue.

When the first such I/O error occurs, FDRPAS puts a non-scrollable message FDR210 on the console to document that it is in the retry loop. IBM I/O error messages are suppressed for the retry I/Os so they appear only for the original error. Message FDR210 appears again when the error is successfully corrected.

NORETRY – An I/O error on the target device while copying tracks immediately terminates the swap. If the I/O error is later corrected, the swap must be restarted.

Default: NORETRY.

VOLSORT=

NO – FDRPAS selects the source volumes in the reverse order that the UCBs appear on the UCB chain, which usually means descending order by device number.

YES – Causes FDRPAS to sort the source volumes by the last digit of the device number (unit address). In most installations, this has the effect of spreading out the SWAP tasks across the LCUs.

Default: NO. FDRPAS selects the source volumes in the reverse order that the UCBs appear on the UCB chain.

NOTE: The order of the MOUNT statements has no influence on the order that FDRPAS selects source volumes.

WTOR=

NO – The FDRW01 and FDRW68 messages are issued as WTO messages, not WTOR messages, so that the operator cannot reply to them. The MSG option of the FDRPAS ISPF panels must be used to reply to these messages (see Section 310.31 "FDRPAS ISPF Interface"). `WTOR=NO` may be appropriate when the system operator is not involved in the FDRPAS swaps, so that inappropriate responses are not made by the operator.

YES – Certain FDRPAS messages requiring a response are issued as WTOR messages so that the system operator can reply to the messages. These messages currently include FDRW01, issued only if `CONFMESS=YES` is specified, and FDRW68, issued for various error conditions. Replies to these messages can be made by the system operator, or by the MSG option of the FDRPAS ISPF panels.

Default: YES.

310.3 SWAP TASK MOUNT STATEMENT

SWAP MOUNT STATEMENT SYNTAX

FDRPAS SWAP task MOUNT statement syntax:

```
MOUNT     VOL=volser                ,SWAPUNIT=uuuu
          ,JESVOL=YES
```

SWAP MOUNT STATEMENT

The SWAP task MOUNT statement follows the `SWAP TYPE=FULL` statement and specifies an online volume to be swapped. One or more MOUNT statements are required, specifying the volumes to be swapped by this SWAP task.

If `MAXTASKS=nn` is specified on the SWAP or SWAPDUMP statement, and multiple MOUNT statements are provided, those MOUNT statements are processed concurrently, up to the MAXTASKS=*nn* limit. If the number of MOUNT statements exceed "*nn*", FDRPAS processes the first "*nn*" MOUNT statements concurrently, and selects the next MOUNT statement as each swap ends.

If `MAXTASKS=nn` is not specified, the MOUNT statements are processed serially, one at a time.

SWAP MOUNT STATEMENT OPERANDS

VOL=

***volser* –** Specifies the volume serial number of an online volume to be swapped.

JESVOL=

On JES2 1.7 or higher or JES3 running on z/OS 1.7 or higher, FDRPAS can automatically determine whether a volume is an active JES volume or a coupling facility volume. When an active JES volume is identified, FDRPAS internally serializes any SYSPRINT or LOG messages to the active JES volumes from this address space while FDRPAS is holding the IOSLEVEL. Because of this, active JES volumes can be swapped with any other volumes or other JES volumes.

YES – Used for systems where FDRPAS cannot automatically identify active JES volumes or coupling facility volumes to inform the FDRPAS job that the device is an active JES volume or a coupling facility volume that requires this special active JES volume handling.

NOTE: This operand is only needed when ANY of the LPARs in the sysplex have a z/OS level lower than z/OS 1.7 or the JES volumes or coupling facility volumes cannot be identified automatically.

SWAPUNIT=

SU=

***uuuu* –** Specifies the z/OS device address of the target device, the new DASD unit that the source volume is swapped to. This must be specified as a 4-digit address, with a leading zero if required. This device should be offline on all systems that are involved in the SWAP and should not contain any needed data (`CHECKTARGET=YES` can be used to ensure that the target device is empty).

It is possible to specify an asterisk (*) for any digit of the `SWAPUNIT=` address. FDRPAS substitutes the corresponding digit of the z/OS address of the source volume (the volume specified by `VOL=`). For example, if the z/OS address of volume PROD01 is 03A4:

```
MOUNT VOL=PROD01,SWAPUNIT=17C*   swaps to device 17C4
MOUNT VOL=PROD01,SWAPUNIT=17*2   swaps to device 17A2
MOUNT VOL=PROD01,SWAPUNIT=17**   swaps to device 17A4
```

This feature allows you to easily swap a set of volumes on adjacent addresses (a "string" of DASD volumes) to a set of adjacent addresses in the new DASD subsystem.

310.4 SWAP TASK EXCLUDE STATEMENT

SWAP EXCLUDE STATEMENT SYNTAX

FDRPAS SWAP task EXCLUDE statement syntax:

EXCLUDE X	CPUID=*cpuid*

SWAP EXCLUDE STATEMENT

The SWAP task EXCLUDE statement optionally follows the `SWAP TYPE=FULL` statement and specifies the hardware CPU serial number of a system image where all source volumes to be processed in this jobstep are known to be offline or not in the I/O configuration. One or more EXCLUDE statements can be specified.

The EXCLUDE statement is ignored when the source volume resides on a storage subsystem on which FDRPAS can determine which LPARs have the volume online. This includes most EMC Symmetrix subsystems, and IBM DS8*xxx* subsystems with the Query Host Access feature. See "Query Host Access Support" in Section 320.1 and "Multi-System Determination" in Section 320.1 for more details. If you feel that you have a need to use an EXCLUDE `CPUID=` statement on such a subsystem, please contact INNOVATION Technical Support.

You should use the EXCLUDE statement only when both of these are true:

- The source volume is in a DASD subsystem where FDRPAS can determine the CPU IDs of the LPARs that have access to the subsystem, but cannot determine which LPARs have the volume online. This includes the IBM 3990-6, IBM 2105 / IBM 2107 / DS6000 / DS8000, and other subsystems that emulate one of these IBM systems, except for IBM DS8*xxx* subsystems with the Query Host Access feature (QHA; discussed above). This does not include most EMC Symmetrix subsystems. On most EMC Symmetrix subsystems, and on IBM subsystems with QHA, FDRPAS can determine which LPARs have the volume online, so EXCLUDE statements are ignored.
- One or more systems have the source DASD volume in their I/O configuration, but you cannot run an FDRPAS MONITOR on those systems. Possible reasons include: non-z/OS operating system (such as Linux, or z/VM not running an FDRPASVM monitor), z/OS LPARs that are not running but not deactivated.

The EXCLUDE statement is used to tell FDRPAS that those systems do not have to participate in the swap.

If you do not use EXCLUDE statements, then FDRPAS may issue the message FDRW68 to indicate that some expected systems are not running FDRPAS MONITOR tasks. Reply "**NO**" to that message so that you can investigate and correct the missing systems.

SWAP EXCLUDE STATEMENT OPERANDS

CPUID=

cpuid **–** Specifies the 10-character CPU serial number of a system image that does not need to participate in the swap.

The CPUID value is the 10-character CPU serial number of a system image. When you run a SIMSWAP job, the CPUID values display on the FDR233 message as shown in this example:

```
FDR303  CARD IMAGE --   SIMSWAP  TYPE=FULL
FDR303  CARD IMAGE --   MOUNT    VOL=HI17C2,SWAPUNIT=17C1
FDR233  CPU WITH (SERIAL# 0212342818) IS ATTACHED TO VOL=HI17C2            - HTC 2107900 TO HTC 2107900
FDR233  CPU WITH (SERIAL# 0112342818) IS ATTACHED TO VOL=HI17C2            - HTC 2107900 TO HTC 2107900
FDR233  CPU WITH (SERIAL# 0512342818) IS ATTACHED TO VOL=HI17C2            - HTC 2107900 TO HTC 2107900
FDR233  CPU WITH (SERIAL# 0912342818) IS ATTACHED TO VOL=HI17C2            - HTC 2107900 TO HTC 2107900
FDRW66  SWAP OF VOL=HI17C2 TO UNIT=17C1 NEEDS TO BE STARTED ON   4 SYSTEMS
```

You can also get the CPUID value is the on a specific z/OS system by executing this console command from a console attached to that system:

```
D M=CPU
```

You receive a response similar to:

D M=CPU

```
IEE174I 15.34.53 DISPLAY M 899
PROCESSOR STATUS
ID  CPU                 SERIAL
00  +                    0212342818
01  +                    1212342818
```

NOTE: The first digit may be non-zero if you have a multi-processor system, as shown in this example. Always make the first digit zero when specifying `CPUID=`. For example, `CPUID=0212342818`.

However, on a z990 system (last four digits are 2084) or any successor system, the first digit may be a legitimate part of the CPUID and should not be zeroed.

310.5 MONITOR SWAP STATEMENT

MONITOR SWAP STATEMENT SYNTAX

FDRPAS MONITOR task SWAP statement syntax:

```
MONITOR   TYPE=SWAP
          ,ALLOWPAV=NO|                ,MAXTASKS=64|
                      YES                         nn
          ,CANCELPROT=NO|              ,PRINT=ALL
                       YES             ,SRVCLASS=classname|
          ,DURATION=nnnn                         HIGHEST>|
          ,DYNMON=NO|                            NULL
                  YES                  ,SWAPDELAY=nnn
          ,LOGMESS=NO|
                   YES
```

MONITOR SWAP STATEMENT

This statement initiates an FDRPAS MONITOR task that monitors for FDRPAS SWAP tasks beginning a SWAP or SWAPDUMP operation. It must be the first statement in the input; only one MONITOR statement is allowed per execution.

A `MONITOR TYPE=SWAP` statement that specifies `DYNMON=NO` must be followed by one or more MOUNT statement to identify the offline devices to be monitored. A `MONITOR TYPE=SWAP` statement that specifies `DYNMON=YES` can specify MOUNT statements, but they are not required. A MONITOR task must be executed **on every system** (other than the SWAP task) that is involved in the swap of an online volume to one of those offline volumes; however, the MONITOR task does not actually participate on the system where the SWAP task is running. If the target device is not offline on an LPAR that a MONITOR task is running, special checking is done by the monitor to ensure that this device is the same target device as specified by the main FDRPAS process and that the device is inactive on the LPAR this monitor is running on. If so, then this volume is varied offline on by this MONITOR task; otherwise, it is left online and fails the operation.

If a MONITOR task with `DYNMON=NO` is monitoring only one offline device (as specified by the following MOUNT statement), then all monitoring is done by the SWAP task; no external or internal tasks are started. Also, FDRPAS does not add SWAPUNITs dynamically. In addition, if any unit is specified, the MONITOR task terminates after the SWAP or SIMSWAPMON completes.

If the MOUNT statement specifies more than one offline device, then this MONITOR task monitors all of those devices, but if it detects that a swap may be beginning on one of them, it starts an internal subtask or external started task to participate in the swap.

If `MAXTASKS=nn` is specified (or defaulted to 64), then up to "*nn*" active swaps are processed as internal subtasks. If the number of monitored devices actively participating in swaps exceeds the value specified or defaulted for `MAXTASKS=`, the additional monitor functions are processed as external started tasks. FDRPAS uses the IBM ASCRE function to create a new started task address space for each MONITOR task, invoking the PASPROC cataloged procedure (see Section 380.16 "FDRPAS and FDRERASE Options (Panel A.I.4.14)").

Since these external MONITOR tasks are separate address spaces, they count against the maximum number of address spaces allowed by the `MAXUSER=` parameter in the IEASYS*xx* PARMLIB member used during the IPL of your z/OS system. If a large number of such external MONITOR tasks are started, some may fail if MAXUSER is exceeded, and it may affect the ability to start other tasks or users.

One or more MONITOR tasks can be submitted by the SWAP task using the PASJOB DD statement.

MONITOR SWAP STATEMENT OPERANDS

TYPE=

Must be specified on the MONITOR statement.

SWAP – Initiates a MONITOR task that monitors offline volumes for a swap operation initiated by a SWAP task.

ALLOWPAV=

The `ALLOWPAV=` operand controls the status of the Parallel Access Volume (PAV) aliases during the swap operation. This operand specifies whether the Parallel Access Volume (PAV) aliases for the source volume are disabled at the start of the swap operation or at the time when the actual swap of the volume occurs. (The PAV aliases for the target volume are always disabled at the start of the SWAP operation.)

NO – Indicates that the Parallel Access Volume (PAV) aliases for the source volume are disabled at the start of the swap operation and enabled after the swap operation completes.

YES – Indicates that the Parallel Access Volume (PAV) aliases for the source volume stay enabled during the initial copy phase and are disabled during the time the actual swap of the volume occurs. This reduces the amount of time that PAV aliases are not available.

Default: NO.

NOTE: ALLOWPAV=YES needs to be specified on all of the SWAP and MONITOR tasks that you wish to leave PAV active.

CANCELPROT=

NO – A `CANCEL (C)` command causes immediate termination of the MONITOR task.

YES – FDRPAS protects this MONITOR task against accidental cancellation. If a `CANCEL (C)` command is issued, it is treated like a `STOP (P)` command. Message FDR267 is issued and the task lets active swaps complete before terminating. You can issue a second `CANCEL (C)` command to force immediate termination.

Default: YES.

DURATION=

***nnnn* –** Specifies the number of idle minutes that the MONITOR task execute; it does not include time that the MONITOR task is actively participating in a swap. The MONITOR task automatically terminates when it has been idle for a total of this many minutes.

Default: The MONITOR task executes until it is terminated by a console `STOP (P)` command, canceled, or until all devices that it is monitoring have been swapped or are online (unless `DYNMON=YES` is coded).

RECOMMENDATION: Contact INNOVATION DATA PROCESSING before using this operand to ensure the correct usage of this operand.

DYNMON=

In a GRS complex or MIM complex environment, FDRPAS MONITOR tasks dynamically add target units that are not in the list regardless of what is specified for the DYNMON= keyword. This keyword specifies how this MONITOR task behaves.

NO – One or more MOUNT control statements must be coded. The MONITOR task terminates once all the devices being monitored have been swapped and are online.

YES – MOUNT control statements are not required (but can be specified) and any devices not in the list are added to the MONITOR task dynamically. The MONITOR task only terminates automatically by control of the DURATION= operand or by the STOP (P) command.

Default: NO.

NOTE: DYNMON=YES cannot be specified for on the MONITOR control statement when SWAP TYPE=TRANSIT is used.

LOGMESS=

NO – No SYSLOG/console messages are written.

YES – Messages are written to SYSLOG (and usually to an operator console) documenting that the swap is occurring, and has completed.

Default: YES.

MAXTASKS=

***nn* –** Specifies the number of internal MONITOR subtasks (0-64) that FDRPAS starts within this address space. If more than these number of volumes are being swapped at one time, the additional MONITOR tasks are started as external started tasks. External subtasks require that the PASPROC cataloged procedure be installed and customized. See the additional explanation above.

Default: 64. Do not override this default unless you receive errors such as insufficient storage errors (80A or 878 ABENDs); this is unlikely to occur.

PRINT=

ALL – Requests additional printout from the MONITOR task, including a detailed list of the tracks updated in each phase.

SRVCLASS=

Allows you to specify the service class for the MONITOR task.

***classname* –** The desired service class.

NOTE: SRVCLASS=SYSTEM is invalid, because FDRPAS is not eligible to be reset into the SYSTEM service class. However, SRVCLASS=SYSSTC is valid.

HIGHEST> – (Note the 8th character is a “greater than” sign.) Set the service class to the same class as the address space with the highest dispatching priority in the system that is X'F8' (248) or less, and that has a single performance period.

NULL – Do not change the service class assigned by WLM.

Default: The default for MONITOR is to not change the service class assigned by WLM, if the service class gives a dispatching priority of X'F0' (240) or higher. If the dispatching priority is less than that, the default is to set the service class to PRDBATHI. If class PRDBATHI is not defined, the default is HIGHEST>.

SWAPDELAY=

nnn **–** Specifies the number of seconds (1-255) that FDRPAS waits between scans of the offline devices it is monitoring to see if an FDRPAS SWAP task has selected one of them as a swap target. The value specified 1/3 or less of the value specified for `SWAPDELAY=` on the SWAP statement in the SWAP task.

Default: 5 seconds (the default on the SWAP statement is 15 seconds). The default should be adequate unless you have a large number of systems participating in the swap process.

310.6 MONITOR SWAP MOUNT STATEMENT

MONITOR SWAP MOUNT STATEMENT SYNTAX

FDRPAS MONITOR task SWAP MOUNT statement syntax:

`MOUNT`	`SWAPUNIT=(`*`uuu1`*`[,`*`uuu2`*`[,…]])`

MONITOR SWAP MOUNT STATEMENT

One or more `MOUNT SWAPUNIT=` statements follow the MONITOR TYPE=SWAP statement. Each specifies an offline target device or set of devices to be monitored to see if one or more of them is selected as the target of a swap by an FDRPAS SWAP task on another system image.

The MONITOR task automatically terminates if it detects that all the devices you have specified on the MOUNT statement have been swapped or are no longer offline unless `DYNMON=YES` is specified on the `MONITOR TYPE=SWAP` statement. With dynamic monitoring, the devices to be monitored are obtained from the SWAP, SWAPDUMP, or SIMSWAPMON task.

MONITOR SWAP MOUNT STATEMENT OPERANDS

SWAPUNIT=

***uuuu* –** Specifies the z/OS device addresses of the target devices to be monitored. The address can be specified as a 4-digit (hex) z/OS device address, or it can be specified as 1, 2, or 3 digits with a trailing asterisk (*); in this case all offline z/OS DASD addresses starting with the prefix specified are monitored. To monitor multiple devices or ranges of devices (by prefix), specify them in parentheses, separated by commas, but you cannot specify more than 255 subparameters (device addresses or address prefix ranges) within the parentheses. If you need to specify more than 255 addresses or prefixes, then you can use more than one `MOUNT SWAPUNIT=` statement. The total number of devices to monitor can be up to 8190. Only devices that are offline and accessible at the time the MONITOR task starts are monitored; if a device is placed offline or becomes accessible after the MONITOR task starts, it is not monitored.

NOTE: If the range to be monitored includes devices that are in your I/O configuration but do not really exist (are not in the hardware configuration of the DASD subsystem), you may receive this console message:

`IOS002A` *`dev`*`, NO PATHS AVAILABLE`

for each such device the first time that the MONITOR task is executed after an IPL. The messages can be ignored. They may occur only under certain releases of OS/390.

For example,

`MOUNT SWAPUNIT=17C0`	monitors one offline device 17C0
`MOUNT SWAPUNIT=17C*`	monitors offline devices in the range 17C0-17CF
`MOUNT SWAPUNIT=17*`	monitors offline devices in the range 1700-17FF
`MOUNT SWAPUNIT=(17*,18*,19A*)`	monitors offline devices in the ranges 1700-17FF, 1800-18FF, and 19A0-19AF

310.7 MONITOR CONFIRM STATEMENT

MONITOR CONFIRM STATEMENT SYNTAX

FDRPAS MONITOR task CONFIRM statement syntax:

```
MONITOR    TYPE=CONFIRMSWAP|
                CONFIRMSPLIT
           ,CONFMESS=NO|                ,MAXCARDS=250|
                     YES                           nnnn

           ,DURATION=nnnn
```

MONITOR CONFIRM STATEMENT

This statement initiates an FDRPAS MONITOR task that monitors for FDRPAS operations that specify `CONFIRMSWAP=YES` or `CONFIRMSPLIT=YES`. It must be the first statement in the input; only one MONITOR statement is allowed per execution.

A `MONITOR TYPE=CONFIRMSWAP` or `TYPE=CONFIRMSPLIT` statement must be followed by one or more MOUNT statements to identify the volumes whose progress is to be monitored. This FDRPAS job may be executed on any system that is involved in the swap of all of the volumes specified.

As soon as all the volumes specified are synchronized, the MONITOR task signals the SWAP task to complete the SWAP or SWAPDUMP of those volumes, all at once. If you submit this job before all of the volumes are ready to complete, it waits for all the volumes to enter that state. If you submit the job after all the volumes are ready to complete, it confirms the operation immediately. (Note that, depending on update activity, a given volume may enter the "ready to complete" state and then exit that state for a few seconds; the MONITOR task waits until all the volumes you specified are ready, which may take a few seconds to a few minutes.)

RECOMMENDATION: *The default of `CONFIRMSWAP=NO` is recommended for a swap operation unless you have a specific reason for wanting a number of DASD volumes to complete their swap at the same time.*

MONITOR CONFIRM STATEMENT OPERANDS

TYPE=

Must be specified on the MONITOR statement.

CONFIRMSPLIT – Initiates a MONITOR task that monitors the operation of a SWAPDUMP task for one or more volumes, and that automatically confirms and completes the FDRPAS operation on those volumes when all the volumes are synchronized and ready to complete.

CONFIRMSWAP – Initiates a MONITOR task that monitors the operation of a SWAP task for one or more volumes, and that automatically confirms and completes the FDRPAS operation on those volumes when all the volume are synchronized and ready to complete.

CONFMESS=

NO – FDRPAS confirms the volumes automatically as soon as they are all synchronized, without operator intervention.

YES – When all volumes specified in this CONFIRM step are synchronized, FDRPAS issues WTOR message FDRW07 to the system operator. If you reply "YES" to the message, then FDRPAS confirms the volumes. If you reply "NO", this CONFIRM step terminates but the swaps are still active. This may be useful when a system automation product is used; it can wait for the FDRW07 message to know that the volumes are synchronized, perform other operations (such as quiescing some applications), and then reply to confirm the volumes.

Default: NO.

DURATION=

nnnn **–** Specifies the number of minutes that the CONFIRMSWAP task keeps checking to see whether the SWAP or SWAPDUMP operation has been initiated for every volume identified on the MOUNT statements. If all volumes have not been initiated within this period, the CONFIRMSWAP fails. If the CONFIRMSWAP task is submitted at the same time as the SWAP or SWAPDUMP jobs, then, depending on the number of volumes and systems involved, it may take anywhere from a few seconds to a minute or two for all of the SWAP or SWAPDUMP tasks to start. The default of 2 should be sufficient for all but extreme cases. If the CONFIRMSWAP fails, the reason is probably that one or more of the volumes specified in the CONFIRMSWAP step have been omitted from the SWAP or SWAPDUMP jobs; be sure to check for this before increasing the DURATION value.

Default: 2.

MAXCARDS=

nnnn **–** Specifies the maximum number of MOUNT statements that can be present in this FDRPAS step, from 1 to 9999.

NOTE: FDRPAS acquires a table with a size of (160 * MAXCARDS) bytes in below-the-line storage, so very large values may cause GETMAIN failures. However, values up to 3000 should not be a problem in FDRPAS.

Default: 250.

310.8 MONITOR CONFIRM MOUNT STATEMENT

MONITOR CONFIRM MOUNT STATEMENT SYNTAX

FDRPAS MONITOR task CONFIRM MOUNT statement syntax:

`MOUNT`	`VOL=`*`volser`*

MONITOR CONFIRM MOUNT STATEMENT

The MONITOR task `MOUNT VOL=` statement follows the `MONITOR TYPE=CONFIRMSWAP` or `CONFIRMSPLIT` statement and specifies a volume to be monitored until it is synchronized with its offline target device. You can specify multiple MOUNT statements, each specifying a volume to be monitored. When **all** volumes specified are synchronized, the MONITOR task signals the SWAP task to confirm the swap and complete the SWAP or SWAPDUMP operation for all the volumes, at one time.

MONITOR CONFIRM MOUNT STATEMENT OPERANDS

VOL=

volser **–** Specifies the complete volume serial number of one online volume to be monitored. A SWAP or SWAPDUMP of the volume must be in progress by an FDRPAS SWAP task.

310.9 MONITOR WAITSWAPTERM STATEMENT

MONITOR WAITSWAPTERM STATEMENT SYNTAX

FDRPAS MONITOR task WAITSWAPTERM statement syntax:

```
MONITOR    TYPE=WAITSWAPTERM
```

MONITOR WAITSWAPTERM STATEMENT

This statement initiates an FDRPAS MONITOR task that monitors the specified volumes to ensure that all the SWAP operations have completed. This MONITOR task does not end until all the volumes have completed the SWAP process.

A `MONITOR TYPE=WAITSWAPTERM` statement must be followed by one or more MOUNT statements to identify the volumes whose progress is to be monitored. This FDRPAS job may be executed on any system that is involved in the swap of all of the volumes specified.

310.10 MONITOR WAITSWAPTERM MOUNT STATEMENT

MONITOR WAITSWAPTERM MOUNT STATEMENT SYNTAX

FDRPAS MONITOR task WAITSWAPTERM MOUNT statement syntax:

`MOUNT`	`VOL=`*`volser`*

MONITOR WAITSWAPTERM MOUNT STATEMENT

The MONITOR task `MOUNT VOL=` statement follows the `MONITOR TYPE=WAITSWAPTERM` statement and specifies a volume to be monitored until the SWAP task has completed. You can specify multiple MOUNT statements, each specifying a volume to be monitored. When **all** volumes specified are complete, the MONITOR task completes and ends.

MONITOR WAITSWAPTERM MOUNT STATEMENT OPERANDS

VOL=

volser **–** Specifies the complete volume serial number of one online volume to be monitored.

310.11 HISTORY STATEMENT

HISTORY STATEMENT SYNTAX

FDRPAS HISTORY statement syntax:

```
HISTORY   TYPE=SWAP
          ,MAXSWAPHISTORY=nnn        ,MAXSWAPHISTORYDAYS=nnnnn
```

HISTORY STATEMENT

The HISTORY statement causes FDRPAS to print the history records for the DASD volumes specified by the MOUNT statements that follow, showing the date each volume was swapped and the source and target devices.

Optional: If `MAXSWAPHISTORY=` and/or `MAXSWAPHISTORYDAYS=` is specified, FDRPAS automatically purges obsolete history records based on the number of times a volume has been swapped, or the number of days since the swap occurred. If both operands are omitted, then the HISTORY statement only prints history records and does not purge any records.

HISTORY STATEMENT OPERANDS

TYPE=SWAP

Must be specified on the HISTORY statement.

MAXSWAPHISTORY=

nnn **–** Specifies the maximum number of history records (1-255) to retain for each selected DASD volume. History records in excess of this number are purged (uncataloged). If a volume was swapped on multiple systems, it retains this many records per system.

MAXSWAPHISTORYDAYS=

nnnnn **–** Specifies the maximum number of days (1-65535) to retain history records for each selected DASD volume. History records created prior to the date calculated from this value are purged (uncataloged).

310.12 HISTORY MOUNT STATEMENT

HISTORY MOUNT STATEMENT SYNTAX

FDRPAS HISTORY MOUNT statement syntax:

`MOUNT`	`VOL=`*`volser`*

HISTORY MOUNT STATEMENT

The `HISTORY MOUNT VOL=` statement follows the `HISTORY TYPE=SWAP` statement and specifies a volume serial or volume serial prefix for which history records are to be displayed and optionally purged. You can specify multiple MOUNT statements, each specifying a volume or group of volumes.

HISTORY MOUNT STATEMENT OPERANDS

VOL=

volser **–** Specifies a volume serial number or volume serial number prefix for volumes for which history records are to be processed. You can specify:

`VOL=`*`volser`*	`a complete volume serial`
`VOL=`*`vol`*`*`	`a prefix, 1-5 characters followed by *`
`VOL=*`	`all volumes with FDRPAS history records`

310.13 SWAPBUILDIX AND EXPANDVTOC STATEMENT

SWAPBUILDIX AND EXPANDVTOC STATEMENT SYNTAX

FDRPAS SWAPBUILDIX and EXPANDVTOC statement syntax:

```
SWAPBUILDIX | TYPE=FULL
EXPANDVTOC  |
            |
            |,LOGMESS=NO|
            |          YES
```

SWAPBUILDIX STATEMENT

This statement initiates an FDRPAS utility task to build or rebuild the indexed VTOC (VTOCIX) on an online volume. It must be the first statement in the input; only one SWAPBUILDIX statement is allowed per execution. SWAPBUILDIX must be followed by one or more MOUNT statements to identify the online volumes on which to build the VTOCIX and may optionally be followed by one or more EXCLUDE statements to exclude certain systems that do not have the volumes online; a maximum of 250 MOUNT/EXCLUDE statements may be present.

The MOUNT and optional EXCLUDE statements that follow the SWAPBUILDIX statement are the same as those used with the SWAP and SWAPDUMP statements (Section 310.3 "SWAP Task MOUNT Statement" and Section 310.4 "SWAP Task EXCLUDE Statement"). On the MOUNT statement, the VOL= operand identifies the volume where the VTOCIX is to be built, and the `SWAPUNIT=` operand identifies an offline monitor DASD volume that FDRPAS uses to coordinate the BUILDIX between the systems. If you have multiple MOUNT statements in one SWAPBUILDIX step, they can use the same `SWAPUNIT=` value since they are executed serially. However, if you submit multiple SWAPBUILDIX jobs for concurrent execution, each one must use a unique offline `SWAPUNIT=` device.

Like FDRPAS SWAP and SWAPDUMP operations, the SWAPBUILDIX task executes on only one system in a multi-system environment. On the other systems, you must execute FDRPAS MONITOR tasks, similar to the MONITOR tasks used with SWAP and SWAPDUMP. The MONITOR tasks must monitor the same offline monitor DASD device(s) specified by `SWAPUNIT=` in the SWAPBUILDIX task. The considerations for running MONITOR tasks and the systems where they must be executed are the same as for SWAP and SWAPDUMP and are discussed elsewhere in this manual, with one exception. A MONITOR task that monitors SWAPBUILDIX or EXPANDVTOC operations, and uses only one SWAPUNIT, defaults to `DURATION=1` after the first operation. That is, once the MONITOR task has completed one SWAPBUILDIX or EXPANDVTOC, the MONITOR task terminates automatically after 1 minute of inactivity. This is appropriate if all of the SWAPBUILDIX or EXPANDVTOC operations are in one job step, or in several steps or jobs that run in quick succession. If there are longer intervals between the SWAPBUILDIX or EXPANDVTOC operations and you want the MONITOR task to stay up between them, specify a sufficient value for `DURATION=`, for example, `DURATION=1000`; then when you are done, terminate the MONITOR with the console command "P *jobname*".

The online volumes must contain a SYS1.VTOCIX.*volser* data set before the SWAPBUILDIX begins; FDRPAS does allocate the VTOCIX if it does not exist. The operation fails if the VTOCIX data set is not present. The IBM *Device Support Facilities (ICKDSF) User's Guide and Reference* (GC35-0033) contains information on allocating and sizing the VTOCIX.

If the VTOCIX is present but inactive on the volume, FDRPAS invokes ICKDSF to build the VTOCIX, then the MONITOR tasks on the other systems update their respective systems with the VTOCIX information. If the VTOCIX is already active, ICKDSF is not invoked, but FDRPAS ensures that the VTOCIX information is correct on every system.

NOTE: There is a small chance that the VTOCIX is disabled again as soon as FDRPAS enables it. This can occur if software on one system (such as IEHLIST or DFSMShsm) accesses the Format 4 DSCB in the VTOC (where the VTOCIX-active flag is stored) before FDRPAS can update the system control blocks to say that the VTOCIX is active. In this case, just run the SWAPBUILDIX again.

SWAPBUILDIX uses code and techniques normally used for a SWAPDUMP operation; you may notice that many FDRPAS messages say "DUMP" when a SWAPBUILDIX is executed. SWAPBUILDIX modifies the offline monitor DASD device, so you must be sure that offline DASD volume does not contain any data that must be preserved. If you want to bring the offline monitor DASD volume online, you need to re-initialize it with ICKDSF first.

A RACROUTE call is issued to verify that the user has READ authority to resource FDRPAS.SWAPBUILDIX in the FACILITY class, if that resource is protected. If FDRPAS.SWAPBUILDIX is not protected, the operation continues.

If you have security checking enabled (see "ALLCALL" in Section 380.14), SWAPBUILDIX checks that the user has ALTER authority to the source volume under the DASDVOL security class. If the user does not have DASDVOL authority, the operation fails. If the volume is not protected by DASDVOL, FDRPAS checks that the user has ALTER authority in the DATASET class to every data set on the volume; if any data set is not authorized, the operation fails.

If you have restricted the use of ICKDSF, then the FDRPAS SWAPBUILDIX job must be authorized to use it.

EXPANDVTOC STATEMENT

The EXPANDVTOC statement is similar in some ways to the SWAPBUILDIX statement, but it has the additional function of expanding the VTOC on the indicated volumes. EXPANDVTOC works only if the program library contains a licensed copy of both FDRPAS and FDRMOVE.

The MOUNT statement used with EXPANDVTOC is described in Section 310.4 "SWAP Task EXCLUDE Statement", except that it supports an additional operand, VTOCSIZE=*nnnnn* (new VTOC size in tracks). If the current VTOC is equal to or larger than VTOCSIZE=, the volume is not changed. For example,

```
MOUNT VOL=volser,SWAPUNIT=uuuu,VTOCSIZE=nnnn
```

The SWAPUNIT is used only for communication between the LPARs; only control information is written to it. You should not run more than one EXPANDVTOC function concurrently. To expand the VTOC, all data sets that follow the VTOC in the space required must be moved. To do so, FDRPAS invokes enhanced functions of FDRCPK (FAST COMPAKTION). EXPANDVTOC moves the VTOCIX and VVDS if necessary, even if they are active, and any other data sets that occupy the required space. However, if any of those other data sets are active (enqueued) they cannot be moved and the EXPANDVTOC fails (messages identify such data sets so that you can take action to make them idle).

The VTOCIX is expanded to be 1/16th of the size of the new VTOC, unless it is already larger than that. This is based on IBM manual *Device Support Facilities (ICKDSF) User's Guide and Reference* (GC35-0033) that documents the max VTOC and INDEX sizes for each size device. On a 3390-9, the maximum VTOC size is 2,944 tracks and the Index size is 150 tracks. The Index comes out as five percent of the VTOC. We default to 6.25 percent. A SWAPBUILDIX is done automatically after building the new VTOC and VTOCIX.

You should pick an appropriate VTOC size based on the expected usage of the volume. Maximum size VTOCs are not recommended unless you expect many 1-track data sets.

EXPANDVTOC holds an enqueue/reserve on the VTOC and VVDS during the expansion. Unless these reserves are converted to global enqueues, this means that all access to the volume from other LPARs is inhibited during the expansion. The elapsed time can be from ten seconds to several minutes. Therefore, you should run the EXPANDVTOC from the LPAR with the most activity to the volume, since that LPAR is still able to access the volume. You may want to run EXPANDVTOC during a quiet time.

If you have third-party products that access the VTOC or VVDS without an enqueue, the third-party products may experience errors if they run during the VTOC expansion.

NOTE: Review "Unmovable Table" in Section 325.11 and "System Volumes" in Section 325.11 for information on data sets that may be active without a SYSDSN enqueue and should be excluded from the EXPANDVTOC process.

Like SWAPBUILDIX, the EXPANDVTOC task executes on only one system in a multi-system environment. On the other systems, you must execute FDRPAS MONITOR tasks, similar to the MONITOR tasks used with SWAP and SWAPDUMP. The MONITOR tasks must monitor the same offline monitor DASD device(s) specified by `SWAPUNIT=` in the EXPANDVTOC task. These MONITOR tasks also update the new VTOC, VTOCIX, and VVDS information on each LPAR. You can use the same SWAPUNIT= on each MOUNT in the EXPANDVTOC step. It is essential the MONITOR tasks run from the same program library as the FDRPAS job, since code updates are required. The considerations for running MONITOR tasks and the systems where they must be executed are the same as for SWAPBUILDIX, (see "SWAPBUILDIX Statement" in Section 310.13).

You should not run multiple concurrent EXPANDVTOC jobs; but if you do, you must use a different SWAPUNIT= for each job.

EXPANDVTOC can be executed either before or after an FDRPAS swap of a volume but not during a swap. If there is not sufficient free space on the volume to expand the VTOC before the swap, you can run it after the swap to a larger volume. `VTOCSIZE=` cannot be specified in a swap operation.

SWAPBUILDIX AND EXPANDVTOC STATEMENT OPERANDS

TYPE=FULL

Must be specified on the SWAPBUILDIX and EXPANDVTOC statement.

#SYSTEMS=

NSYSTEMS=

NOTE: This operand is no longer documented and INNOVATION DATA PROCESSING recommends that it NOT be used unless you are swapping a really old control unit. Contact INNOVATION DATA PROCESSING if you receive an FDR234 REASON=R message indicating that you need to specify `#SYSTEMS=`.

LOGMESS=

NO – No SYSLOG/console messages are written.

YES – Messages are written to SYSLOG (and usually to an operator console) documenting that the SWAPBUILDIX is occurring, and has completed.

Default: NO

MIN#SYSTEMS=

MINNSYSTEMS=

NOTE: This operand is no longer documented and INNOVATION DATA PROCESSING recommends that it NOT be used. The `EXCLUDE CPUID=` statement (see Section 310.4 "SWAP Task EXCLUDE Statement") is the recommended way of handling systems or LPARs that are connected to the source DASD device, but cannot bring that device online.

310.14 MONITOR RESET AND VARY STATEMENT

MONITOR RESET AND VARY STATEMENT SYNTAX

FDRPAS MONITOR task VARY statement syntax:

```
MONITOR     TYPE=RESETVOL|
                 VARYONLINE
```

MONITOR RESET AND VARY STATEMENT

This type of MONITOR statement resets the label and optionally varies online an offline DASD device that was:

- The original source volume for a volume that was successfully swapped to a new device.
- The target device of a successfully completed SWAPDUMP operation.
- The target device of an unsuccessful SWAP or SWAPDUMP (one that began but failed or was canceled before it completed).

All these offline DASD devices have a volume label that has been modified so that they cannot be varied online. This function resets the volume label so that it can again be varied online, and also modifies the volume serial so that it can be mounted. All existing data on the volume is preserved. Note that if the target device of an unsuccessful operation is reset, the data on that device may not be complete or may not be usable.

The MONITOR statement is followed by one or more MOUNT statements (see Section 310.15 "RESET and VARY MOUNT Statement") to identify the offline DASD devices to be processed.

Although the statement name is MONITOR, this does not initiate an FDRPAS MONITOR task and it does not participate in swaps. See Section 310.27 "VARY Examples" for examples.

NOTE: This function replaces the FDRPASV1 utility, which was used in earlier releases to correct the volume label but did not change the volume serial number or vary the device online. FDRPASV1 can still be used but is no longer documented.

MONITOR RESET AND VARY STATEMENT OPERANDS

TYPE=

Must be specified on the MONITOR statement.

RESETVOL – Resets the volume label and changes the volume serial but does not vary the volume online. This is generally used when you plan to vary the volume online to another system.

VARYONLINE – Resets the volume label, changes the volume serial, and varies online the DASD volume(s) that are specified by following MOUNT statements. It is varied online only on the system where this FDRPAS job executes; on other systems you must issue a console VARY command if you need to access it.

310.15 RESET AND VARY MOUNT STATEMENT

RESET AND VARY MOUNT STATEMENT SYNTAX

FDRPAS MONITOR task VARY MOUNT statement syntax:

MOUNT	VARYUNIT=*uuuu*	,NVOL=*vvvvvv*

RESET AND VARY MOUNT STATEMENT

The monitor MOUNT VARYUNIT= statement follows the MONITOR TYPE=VARYONLINE or RESETVOL statement and specifies an offline DASD device or a range of devices whose volume label is to be reset and optionally varied online. You can specify multiple MOUNT statements, each specifying a different device to be monitored.

The device addresses specified must:

- Be offline DASD devices.
- Have a volume label that indicates they were the original source volume in a successful swap, the target devices in a successful SWAPDUMP, or the target device in an unsuccessful SWAP or SWAPDUMP.

RESET AND VARY MOUNT STATEMENT OPERANDS

VARYUNIT=

***uuuu* –** Specifies the 4-digit device address of an offline DASD device to be processed. However, trailing digits may be asterisks to indicate a range of devices. For example,

```
VARYUNIT=123*    processes all offline devices in the range
                 1230 to 123F
VARYUNIT=12**    processes all offline devices in the range
                 1200 to 12FF
```

If a range is specified, FDRPAS ignores all nonexistent devices, devices that are online, and those that do not contain an FDRPAS-generated volume label, with no error message; messages are printed only for DASD volumes that FDRPAS has successfully reset. If a specific device address is given, an error message results if it is not in the proper condition to reset.

NVOL=

***vvvvvv* –** Specifies a new volume serial to be assigned to the DASD volume, from 1 to 6 characters; if it is less than 6, the assigned serial has trailing blanks. Normally this is a new serial, since the original DASD volume serial is usually still online.

NVOL= may contain an asterisk (*) in any position. For any character that is an asterisk, the corresponding character from the original volume serial of the DASD volume is preserved. For example, if the original volume serial number was “ABC123”, and NVOL=XYZ*5* is specified, the new serial is “XYZ153”. Note that if VARYUNIT= specifies a range of devices, NVOL= should specify a mask so that duplicate volume serials are not generated.

If the volume serial requested is already online on this system, the volume label is not modified and the request fails with a diagnostic message. If the volume serial is correct, you need to execute this function on another system where that volume is not online.

If you want to retain the original volume serial of the DASD volume, you can either specify the original serial or NVOL=******, but that original serial must not be online on this system.

NOTE: Neither the VTOCIX (SYS1.VTOCIX.*volser*) nor the VVDS (SYS1.VVDS.V*volser*) is renamed. The VTOCIX still works correctly, but the VVDS is not accessible so VSAM clusters on the volume are not usable. Any attempt to allocate new VSAM or SMS-managed data sets to the volume results in the creation of a second VVDS with the new volume serial, which may cause other problems.

310.16 LICENSE STATEMENT

LICENSE STATEMENT SYNTAX

FDRPAS LICENSE statement syntax:

```
LICENSE    TYPE=SWAP                    ,PRINT=ALL
```

LICENSE STATEMENT

The LICENSE statement causes FDRPAS to quickly scan all of the DASD storage that is online on the system where it is executed. It lists the DASD subsystems it finds by manufacturer, serial number, and gigabytes (GB) of online storage in the subsystem. It also shows the equivalent number of 3390-3 volumes in each subsystem. The total GB and 3390-3 equivalents for the entire system are also shown.

If `PRINT=ALL` is specified, it additionally reports on every online DASD volume, showing its device address, size in cylinders and size in GB. This report is written to the TAPE1 DD statement, which is dynamically allocated as SYSOUT=* if it is not present in the JCL.

If a DASD subsystem is listed as type UNKNOWN, this means that FDRPAS was unable to determine the manufacturer and serial number of the subsystem; this can occur on IBM 3990-3 and older control units.

FDRPAS license charges may be based on the total online GB of data on your systems, so INNOVATION DATA PROCESSING may ask you to run a `LICENSE TYPE=SWAP` on every system in order to determine these charges.

LICENSE STATEMENT OPERANDS

TYPE=SWAP

Must be specified on the LICENSE Statement.

PRINT=

ALL – FDRPAS generates a report showing every individual online DASD volume. If omitted, only the totals for each DASD subsystem are printed.

310.20 LICENSE EXAMPLES

FDRPAS license charges may be based on the total online GB of data on your systems, so INNOVATION DATA PROCESSING may ask you to run a LICENSE TYPE=SWAP on every system in order to determine these charges.

All examples in this section are found in the JCL library installed with FDRPAS. The member names are PA31020*x*.

LICENSE REPORT EXAMPLE

FDRPAS displays the amount of online storage in every DASD subsystem attached to this system.

```
//LICENSE  EXEC PGM=FDRPAS,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  LICENSE  TYPE=SWAP
/*
```

This produces an output similar to:

```
FDR303  CARD IMAGE --   LICENSE TYPE=SWAP,PRINT=ALL
CONTROLLER SSID STORGRP  UNIT VOLSER   % ALLOC  # CYLS  SIZE(GB)    TOTAL(GB)  # 3390-3
---------- ---- -------- ---- ------   -------  ------  --------  -----------  --------
 *UNKNOWN*                                 18                        28.647        10
 EMC 01336                                 56                        98.371        34
 IBM 12345                                 12                       225.302        79
 IBM 22935                                 18                       194.225        68
 IBM 70941                                 44                       117.933        41
 IBM 91468                                 24                        15.136         5
 EMC 00938                                 10                        48.458        17
 EMC 01889                                  4                       200.515        70
 HTC 45278                                 19                        63.764        22
 IBM 00000                                 84                         5.676         2
CPU SERIAL SYSTEM NAME                % ALLOC                      TOTAL(GB)  # 3390-3
---------- -----------                -------                     -----------  --------
0209417060 CPUB                           21                       998.032       351
```

LICENSE REPORT WITH VOLUME DETAIL EXAMPLE

FDRPAS displays the amount of online storage in every DASD subsystem attached to this system. Additionally, it dynamically allocates a `TAPE1 DD SYSOUT=*` statement and writes a report to it showing details of every online DASD volume.

```
//LICENSE  EXEC PGM=FDRPAS,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  LICENSE  TYPE=SWAP,PRINT=ALL
/*
```

310.21 FDRPAS SWAP TASK EXAMPLES

These are examples of SWAP operations, where the volumes involved are moved to the new DASD devices specified by the `SWAPUNIT=` operand. The examples first show performing the task by using the ISPF panels. This is followed by a comparable JCL example that can be found in the JCL library installed with FDRPAS. The member names are PA31021*x*. INNOVATION DATA PROCESSING strongly recommends you use the ISPF panels to generate the FDRPAS JCL and control cards.

NOTE: See Section 310.31 "FDRPAS ISPF Interface" for detailed information on using and submitting jobs using the ISPF panels.

SWAP A GROUP OF VOLUMES ON MULTIPLE SYSTEMS EXAMPLE

In this example, the volumes in SMS storage group "PROD9" are swapped to offline units in the 21C* range. To perform that, type "prod9" in the "SMS Storage Group" field and press ENTER.

```
COMMAND ===>                                                  SCROLL ===> PAGE
Menu                                                             Panel: 1 of 3
Command  Volume Unit              SMS   Swap to                  Refresh 0
         Serial Addr        CU  Storage Offline
          Mask  Mask SSID Serial Group   Unit   Status                 11:43:23
-------  ------ ---- ---- ----- -------- ---- --------------------------------
'''                          prod9
```

The information for the devices in SMS storage group PROD9 are displayed. From this panel, type the target mask "21c*" in the "Swap to Offline Unit" field to generate the offline units for the displayed volumes.

```
Menu                                                             Panel: 1 of 3
Command  Volume Unit              SMS   Swap to                  Refresh 0
         Serial Addr        CU  Storage Offline
          Mask  Mask SSID Serial Group   Unit   Status                 11:46:51
-------  ------ ---- ---- ----- -------- ---- --------------------------------
         IDPLB8 8068 1000 31290 PROD9    21c* INACTIVE
'''
         IDPLB7 8067 1000 31290 PROD9         INACTIVE
         IDPLB6 8066 1000 31290 PROD9         INACTIVE
         IDPLB5 8065 1000 31290 PROD9         INACTIVE
         IDPLB4 8064 1000 31290 PROD9         INACTIVE
         IDPLB3 8063 1000 31290 PROD9         INACTIVE
```

The source and target devices are verified on the executing system and additional source volume information is displayed. If no discrepancies are found with the source and target volumes, "INACTIVE" is displayed in the "Status" field.

```
Menu                                                             Panel: 1 of 3
Command  Volume Unit              SMS   Swap to                  Refresh 0
         Serial Addr        CU  Storage Offline
          Mask  Mask SSID Serial Group   Unit   Status                 11:49:16
-------  ------ ---- ---- ----- -------- ---- --------------------------------
         IDPLB8 8068 1000 31290 PROD9    21C8 INACTIVE
         IDPLB7 8067 1000 31290 PROD9    21C7 INACTIVE
         IDPLB6 8066 1000 31290 PROD9    21C6 INACTIVE
         IDPLB5 8065 1000 31290 PROD9    21C5 INACTIVE
         IDPLB4 8064 1000 31290 PROD9    21C4 INACTIVE
         IDPLB3 8063 1000 31290 PROD9    21C3 INACTIVE
```

Once all the information is entered, Use the "SIMSWAP job" option on the "Menu"

drop-down list to generate the job and control cards to perform the SIMSWAP.

FDRPAS PULL-DOWN MENU

```
Active          SWAP job        CLear entries
COnfirm         SWAPDUMP job    RESEt status
Message         CONFIRM job
SUspend         MONITOR job
REsume          SIMSWAP job
ABORT           SIMSWAPMON job
Options         AUTOSWAP job
HIstory         HYPERSWAP job
SOrt

Place the cursor next to the command to
execute and press enter. Press PF3 to exit.
Press PF1 for the Help tutorials.
```

Pressing ENTER displays the generated JCL and control statements that can be modified if needed. Type "submit" on the TSO command line to submit this for execution. The options that are generated are the values from the FDRPAS Options panel (see Section 310.31 "FDRPAS ISPF Interface").

SIMSWAP GENERATED JCL AND CONTROL STATEMENTS

```
Command ===> submit                                          Scroll ===> CSR
****** ***************************** Top of Data ******************************
…
000005 //FDRPAS   EXEC PGM=FDRPAS,REGION=0M
000006 //STEPLIB  DD  DISP=SHR,DSN=fdrpas.loadlib
000007 //SYSPRINT DD  SYSOUT=*
000008 //SYSUDUMP DD  SYSOUT=*
000009 //SYSIN    DD  *
000010   SIMSWAP  TYPE=FULL,
000011         CONFIRMSWAP=NO,
000012         CONFMESS=NO,
000013         LOGMESS=YES,
000014         PACEDELAY=0,
000015         SWAPDELAY=15,
000016         CHECKTARGET=YES,
000017         LARGERSIZE=OK,
000018         PACING=STATIC,
000019         MAXTASKS=1,
000020         MAXACTIVESWAPS=NO,
000021         WTOR=YES
000022   MOUNT VOL=IDPLB8,SWAPUNIT=21C8
000023   MOUNT VOL=IDPLB7,SWAPUNIT=21C7
000024   MOUNT VOL=IDPLB6,SWAPUNIT=21C6
000025   MOUNT VOL=IDPLB5,SWAPUNIT=21C5
000026   MOUNT VOL=IDPLB4,SWAPUNIT=21C4
000027   MOUNT VOL=IDPLB3,SWAPUNIT=21C3
```

The output from this SIMSWAP run includes the CPU serial numbers and the number of systems that need to have MONITOR tasks running.

```
…
FDR233  CPU WITH (SERIAL# 026E0D2096) IS ATTACHED TO VOL=IDPLB3          - EMC 2105F20 TO IBM 2107921
FDR233  CPU WITH (SERIAL# 016E0D2096) IS ATTACHED TO VOL=IDPLB3          - EMC 2105F20 TO IBM 2107921
FDR233  CPU WITH (SERIAL# 096E0D2096) IS ATTACHED TO VOL=IDPLB3          - EMC 2105F20 TO IBM 2107921
FDRW66  SWAP OF VOL=IDPLB3 TO UNIT=21C3 NEEDS TO BE STARTED ON   3 SYSTEMS
…
```

The CPU serial numbers from the SIMSWAP output is needed to create the JCL for the MONITOR tasks. Use the FDRPAS Options panel (go to the pull-down MENU and select "Options") to type the CPU serial numbers and the JCL necessary to route jobs on the other LPARs.

FDRPAS OPTIONS PANEL (A.P.OPTIONS) – ENTER CPUID AND ROUTING JCL

```
                                                                More:   - +
 Job Statement Information:
   ===> //pfxE  JOB 'user id',NOTIFY=pfx,CLASS=A
   ===> //*
   ===> //*
   ===> //*

   STEPLIB    ===> 'fdrpas.loadlib'

 Dataset name containing Swap Exclude statements  (optional)
   Dsname     ===>
   Member     ===>                       (required for partitioned data set)

 Job Statement Information for Monitor Jobs:

 Monitor #1
        CPUID ===> 0912342818            (specify to include MONITOR in PASJOB)
   ===> //PASMONA  JOB (acct),'user id',MSGLEVEL=(1,1),
   ===> //            MSGCLASS=X,CLASS=M
   ===> /*ROUTE    XEQ JESCPUA
   ===>
 Monitor #2
        CPUID ===> 0112342818            (specify to include MONITOR in PASJOB)
   ===> //PASMONB  JOB (acct),'user id',MSGLEVEL=(1,1),
   ===> //            MSGCLASS=X,CLASS=M
   ===> /*ROUTE    XEQ JESCPUC
   ===>
 Monitor #3
        CPUID ===> 0212342818            (specify to include MONITOR in PASJOB)
   ===> //PASMONC  JOB (acct),'user id',MSGLEVEL=(1,1),
   ===> //            MSGCLASS=X,CLASS=M
   ===> /*ROUTE    XEQ JESCPUB
   ===>
…
 Monitor #12
        CPUID ===>                       (specify to include MONITOR in PASJOB)
   ===>
   ===>
   ===>
   ===>
```

The Options panel allows specification of up to 12 MONITOR tasks. If you need to add more than 12 MONITOR tasks, they can be added manually after the JCL is generated.

Use the "SIMSWAPMON *job*" option from the FDRPAS pull-down MENU to generate the JCL and control statements needed to run the SIMSWAPMON job. Type "submit" on the TSO command line to submit this for execution.

SIMSWAPMON GENERATED JCL AND CONTROL STATEMENTS

```
Command ===> submit                                          Scroll ===> CSR
****** ***************************** Top of Data ******************************
…
000005 //FDRPAS   EXEC PGM=FDRPAS,REGION=0M
000006 //STEPLIB  DD  DISP=SHR,DSN=fdrpas.loadlib
000007 //SYSPRINT DD  SYSOUT=*
000008 //SYSUDUMP DD  SYSOUT=*
000009 //SYSIN    DD  *
000010   SIMSWAPMON  TYPE=FULL,
000011         CONFIRMSWAP=NO,
000012         CONFMESS=NO,
000013         LOGMESS=YES,
000014         PACEDELAY=0,
000015         SWAPDELAY=15,
000016         CHECKTARGET=YES,
000017         LARGERSIZE=OK,
000018         PACING=STATIC,
000019         MAXTASKS=1,
000020         MAXACTIVESWAPS=NO,
000021         WTOR=YES
000022   MOUNT VOL=IDPLB8,SWAPUNIT=21C8
000022   MOUNT VOL=IDPLB7,SWAPUNIT=21C7
…
000028 //PASJOB    DD  DATA,DLM=ZZ
000029 //*CPUID=0912342818
000030 //PASMONU  JOB (acct),'user id',MSGLEVEL=(1,1),
000031 //            MSGCLASS=X,CLASS=M
000032 /*ROUTE    XEQ JESCPUA
000034 //FDRPAS   EXEC PGM=FDRPAS,REGION=0M
000035 //STEPLIB  DD  DISP=SHR,DSN=fdrpas.loadlib
000036 //SYSPRINT DD  SYSOUT=*
000037 //SYSUDUMP DD  SYSOUT=*
000038 //SYSIN    DD  *
000039   MONITOR TYPE=SWAP,
000040           MAXTASKS=64,
000041           DYNMON=YES,
000042           DURATION=100,
000043           LOGMESS=NO,
000044           MAXCARDS=250
000045   MOUNT   SWAPUNIT=(21C8,
000046                     21C7,
000047                     21C6,
000048                     21C5,
000049                     21C4,
000050                     21C3)
000051 //*CPUID=0112342818
000052 //PASMONV  JOB (acct),'user id',MSGLEVEL=(1,1),
000053 //            MSGCLASS=X,CLASS=M
000054 /*ROUTE    XEQ JESCPUC
000056 //FDRPAS   EXEC PGM=FDRPAS,REGION=0M
000057 //STEPLIB  DD  DISP=SHR,DSN=fdrpas.loadlib
000058 //SYSPRINT DD  SYSOUT=*
000059 //SYSUDUMP DD  SYSOUT=*
000060 //SYSIN    DD  *
000061   MONITOR TYPE=SWAP,
…
```

The output from this SIMSWAPMON run shows that CHECKSOURCE=YES was specified "CPK320I" as well as the acknowledgment of the MONITOR tasks from the other LPARs FDR233. It also shows that the job completed successfully FDR999 and is ready for the actual SWAP job to be run.

```
FDR303  CARD IMAGE --   SIMSWAPMON  TYPE=FULL,
FDR303  CARD IMAGE --          CONFIRMSWAP=NO,
FDR303  CARD IMAGE --          CONFMESS=NO,
FDR303  CARD IMAGE --          LOGMESS=YES,
FDR303  CARD IMAGE --          PACEDELAY=0,
FDR303  CARD IMAGE --          SWAPDELAY=15,
FDR303  CARD IMAGE --          CHECKSOURCE=YES,
FDR303  CARD IMAGE --          CHECKTARGET=YES,
FDR303  CARD IMAGE --          LARGERSIZE=OK,
FDR303  CARD IMAGE --          PACING=STATIC,
FDR303  CARD IMAGE --          MAXTASKS=1,
FDR303  CARD IMAGE --          MAXACTIVESWAPS=NO,
FDR303  CARD IMAGE --          WTOR=YES
FDR303  CARD IMAGE --   MOUNT VOL=IDPLB8,SWAPUNIT=21C8
FDR303  CARD IMAGE --   MOUNT VOL=IDPLB7,SWAPUNIT=21C7
FDR303  CARD IMAGE --   MOUNT VOL=IDPLB6,SWAPUNIT=21C6
FDR303  CARD IMAGE --   MOUNT VOL=IDPLB5,SWAPUNIT=21C5
FDR303  CARD IMAGE --   MOUNT VOL=IDPLB4,SWAPUNIT=21C4
FDR303  CARD IMAGE --   MOUNT VOL=IDPLB3,SWAPUNIT=21C3
FDR183  FDR SUBMITTED FDRPAS MONITOR JOBNAME=PASMONU  FOR CPUID=0912342818
FDR183  FDR SUBMITTED FDRPAS MONITOR JOBNAME=PASMONV  FOR CPUID=0112342818
FDR256  --- REPORT FROM SYSPRIN1 ---
CPK320I  COMPAKTOR OPTIONS IN EFFECT -- CHECKSOURCE-------------IDPLB8
CPK529I  INDEXED VTOC EXISTS ON VOLUME=IDPLB8
VOLUME SUMMARY --- DEVICE TYPE ..............................  3390-3
                   NO. OF TRACKS ............................. 50085
                   NO. OF CYLINDERS ...........................  3339
                   NO. OF ALLOCATED TRACKS ..................... 48834
                   PERCENTAGE OF VOLUME IN USE .................    98
CPK600I  COMPAKTOR SUCCESSFULLY COMPLETED CHECKSOURCE RUN ------ VOLUME=IDPLB8
FDR233  CPUB     (SERIAL# 0212342818) ACKNOWLEDGES THE SWAP OF VOL=IDPLB8  - HTC 2107900 TO HTC 2107900
FDR233  CPUC     (SERIAL# 0112342818) ACKNOWLEDGES THE SWAP OF VOL=IDPLB8 AND HAS JOINED IN SWAP OF UNIT=8068 TO 21C8
FDR233  CPUA     (SERIAL# 0912342818) ACKNOWLEDGES THE SWAP OF VOL=IDPLB8 AND HAS JOINED IN SWAP OF UNIT=8068 TO 21C8
FDRW66  SIMSWAP VOL=IDPLB8 TO UNIT=21C8 COMPLETED    3 SYSTEMS (CPUB CPUC CPUA)
…
FDR999   FDR SUCCESSFULLY COMPLETED
```

This listing shows that two MONITOR tasks were started by FDRPAS. Also, since CHECKSOURCE=YES was specified, it shows the COMPAKTOR stats from the volume plus the VTOC and VVDS had no errors. The last part of the messages show the systems that participate in the swap.

After running and verifying that the SIMSWAPMON job completed with no errors, return to the FDRPAS ISPF panels and generate the SWAP job that performs the actual movement of the data. The JCL that is generated looks similar to the SIMSWAPMON JCL. When everything is ready, type “submit” on the TSO command like to submit the JCL.

SWAP GENERATED JCL AND CONTROL STATEMENTS

```
Command ===> submit                                          Scroll ===> CSR
********************************** Top of Data ******************************
…
000005 //FDRPAS   EXEC PGM=FDRPAS,REGION=0M
000006 //STEPLIB  DD  DISP=SHR,DSN=fdrpas.loadlib
000007 //SYSPRINT DD  SYSOUT=*
000008 //SYSUDUMP DD  SYSOUT=*
000009 //SYSIN    DD  *
000010   SWAP  TYPE=FULL,
000011         CONFIRMSWAP=NO,
000012         CONFMESS=NO,
000013         LOGMESS=YES,
000014         PACEDELAY=0,
000015         SWAPDELAY=15,
000016         CHECKTARGET=YES,
000017         LARGERSIZE=OK,
000018         PACING=STATIC,
000019         MAXTASKS=1,
000020         MAXACTIVESWAPS=NO,
000021         WTOR=YES
000022   MOUNT VOL=IDPLB8,SWAPUNIT=21C8
000023   MOUNT VOL=IDPLB7,SWAPUNIT=21C7
…
000028 //PASJOB    DD  DATA,DLM=ZZ
000029 //*CPUID=0912342818
000030 //PASMONG  JOB (acct),'user id',MSGLEVEL=(1,1),
000031 //            MSGCLASS=X,CLASS=M
000032 /*ROUTE    XEQ JESCPUA
000034 //FDRPAS   EXEC PGM=FDRPAS,REGION=0M
000035 //STEPLIB  DD  DISP=SHR,DSN=fdrpas.loadlib
000036 //SYSPRINT DD  SYSOUT=*
000037 //SYSUDUMP DD  SYSOUT=*
000038 //SYSIN    DD  *
000039   MONITOR TYPE=SWAP,
000040           MAXTASKS=64,
000041           DYNMON=YES,
000042           DURATION=100,
000043           LOGMESS=NO,
000044           MAXCARDS=250
000045   MOUNT   SWAPUNIT=(21C8,
000046                     21C7,
000047                     21C6,
000048                     21C5,
000049                     21C4,
000050                     21C3)
000051 //*CPUID=0112342818
000052 //PASMONH  JOB (acct),'user id',MSGLEVEL=(1,1),
000053 //            MSGCLASS=X,CLASS=M
000054 /*ROUTE    XEQ JESCPUC
000056 //FDRPAS   EXEC PGM=FDRPAS,REGION=0M
000057 //STEPLIB  DD  DISP=SHR,DSN=fdrpas.loadlib
000058 //SYSPRINT DD  SYSOUT=*
000059 //SYSUDUMP DD  SYSOUT=*
000060 //SYSIN    DD  *
…
```

SIMULATE A SWAP EXAMPLE

Execute a simulated swap (SIMSWAP statement) to validate the parameters being used for the real SWAP. No MONITOR tasks are required. This displays all of the LPARs with access to the source volumes on DASD subsystems where FDRPAS can identify the attached LPARs.

```
//SIMSWAP  EXEC PGM=FDRPAS,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  SIMSWAP  TYPE=FULL,MAXTASKS=3,LARGERSIZE=OK
  MOUNT    VOL=DATA22,SWAPUNIT=07C3
  MOUNT    VOL=PROD12,SWAPUNIT=425C
/*
```

SIMULATE A SWAP WITH MONITOR TASKS EXAMPLE

Execute a simulated swap with monitor checking (SIMSWAPMON statement) to validate the parameters being used for the real SWAP. MONITOR tasks are required for all LPARs in the sysplex. In this example, one LPAR is a VM system requiring an `EXCLUDE CPUID=` statement to exclude that LPAR. If you are licensed for FDRPASVM (see Chapter 321 "FDRPASVM") an `EXCLUDE` statement is not necessary for supported VM systems and the VM system does not need to be quiesced.

```
//SIMSWAPM EXEC PGM=FDRPAS,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//FDRSUMM  DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  SIMSWAPMON TYPE=FULL,MAXTASKS=4,LARGERSIZE=OK,CHECKSOURCE=YES
  EXCLUDE  CPUID=0512342818
  MOUNT    VOL=HI17CC,SWAPUNIT=17CA
  MOUNT    VOL=HI17C2,SWAPUNIT=17C1
/*
```

The SYSPRINT output from this job is similar to:

```
FDR303  CARD IMAGE --   SIMSWAPMON TYPE=FULL,MAXTASKS=4,LARGERSIZE=OK,CHECKSOURCE=YES
FDR303  CARD IMAGE --   EXCLUDE  CPUID=0512342818
FDR303  CARD IMAGE --   MOUNT    VOL=HI17CC,SWAPUNIT=17CA
FDR303  CARD IMAGE --   MOUNT    VOL=HI17C2,SWAPUNIT=17C1
FDR256  --- REPORT FROM SYSPRIN1 ---
CPK320I  COMPAKTOR OPTIONS IN EFFECT -- CHECKSOURCE------------HI17CC
CPK529I  INDEXED VTOC EXISTS ON VOLUME=HI17CC
VOLUME SUMMARY --- DEVICE TYPE ..............................  3390
                   NO. OF TRACKS ............................. 16695
                   NO. OF CYLINDERS ..........................  1113
                   NO. OF ALLOCATED TRACKS ....................    18
                   PERCENTAGE OF VOLUME IN USE .................     0
CPK600I  COMPAKTOR SUCCESSFULLY COMPLETED CHECKSOURCE RUN ------ VOLUME=HI17CC
FDR233  CPU WITH (SERIAL# 0512342818) WAS EXCLUDED
FDR233  CPUB     (SERIAL# 0212342818) ACKNOWLEDGES THE SWAP OF VOL=HI17CC  - HTC 2107900 TO HTC 2107900
FDR233  CPUC     (SERIAL# 0112342818) ACKNOWLEDGES THE SWAP OF VOL=HI17CC AND HAS JOINED IN SWAP OF UNIT=17CC TO 17CA
FDR233  CPUA     (SERIAL# 0912342818) ACKNOWLEDGES THE SWAP OF VOL=HI17CC AND HAS JOINED IN SWAP OF UNIT=17CC TO 17CA
FDRW66  SIMSWAP VOL=HI17CC TO UNIT=17CA COMPLETED    3 SYSTEMS (CPUB CPUC CPUA)
FDR256  --- REPORT FROM SYSPRIN2 ---
CPK320I  COMPAKTOR OPTIONS IN EFFECT -- CHECKSOURCE------------HI17C2
CPK529I  INDEXED VTOC EXISTS ON VOLUME=HI17C2
VOLUME SUMMARY --- DEVICE TYPE ..............................  3390-3
                   NO. OF TRACKS ............................. 50085
                   NO. OF CYLINDERS ..........................  3339
                   NO. OF ALLOCATED TRACKS ..................... 48834
                   PERCENTAGE OF VOLUME IN USE .................    98
CPK600I  COMPAKTOR SUCCESSFULLY COMPLETED CHECKSOURCE RUN ------ VOLUME=HI17C2
FDR233  CPU WITH (SERIAL# 0512342818) WAS EXCLUDED
FDR233  CPUB     (SERIAL# 0212342818) ACKNOWLEDGES THE SWAP OF VOL=HI17C2  - HTC 2107900 TO HTC 2107900
FDR233  CPUC     (SERIAL# 0112342818) ACKNOWLEDGES THE SWAP OF VOL=HI17C2 AND HAS JOINED IN SWAP OF UNIT=17C2 TO 17C1
FDR233  CPUA     (SERIAL# 0912342818) ACKNOWLEDGES THE SWAP OF VOL=HI17C2 AND HAS JOINED IN SWAP OF UNIT=17C2 TO 17C1
FDRW66  SIMSWAP VOL=HI17C2 TO UNIT=17C1 COMPLETED    3 SYSTEMS (CPUB CPUC CPUA)
FDR999   FDR SUCCESSFULLY COMPLETED
```

The FDRSUMM output from this job is similar to:

VOLSER	COMP CODE	ELAPSED TIME(MIN)	VOLUME SIZE(CYL)	DASD BYTES READ FROM VOL	DATASETS PROCESSED	UPDATED TRACKS	TRACKS SWAPPED
HI17CC	0	0.0	1,113	0	0	0	0
HI17C2	0	0.0	3,339	0	0	0	0

SWAP A VOLUME ON MULTIPLE SYSTEMS EXAMPLE

A volume accessible to multiple system images is swapped to a new device. FDRPAS determines the number of systems that have access to the volume. Prior to running this job, you must start FDRPAS MONITOR tasks on each of the systems, to monitor the target device (see Section 310.23 "MONITOR SWAP Examples"). The target device must be offline on all systems.

```
//SWAP     EXEC PGM=FDRPAS,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//FDRSUMM  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  SWAP     TYPE=FULL
  MOUNT    VOL=DATA22,SWAPUNIT=07C3
/*
```

A partial output from the swap step looks similar to:

```
FDR241  FDRPAS SUCCESSFULLY COMPLETED SWAP OF VOL=DATA22 TO UNIT=07C3 ON CPUB
FDR241  FDRPAS SUCCESSFULLY COMPLETED SWAP OF VOL=DATA22 TO UNIT=07C3 ON CPUA
FDR241  FDRPAS SUCCESSFULLY COMPLETED SWAP OF VOL=DATA22 TO UNIT=07C3 ON CPUC
FDR007  ENDING   TIME OF FULL VOL SWAP -- 16.31.56 -- UNIT=3390-3 ,IN=D#DATA22,OUTPUT=TAPE1
FDR122  OPERATION STATISTICS FOR 3390 VOLUME.................DATA22
FDR122                        CYLINDERS ON VOLUME.............3,339
FDR122                        DATASETS PROCESSED..............234
FDR122                        BYTES READ FROM DASD.......237,893,477
FDR122                        DASD TRACKS SWAPPED.............4,986
FDR122                        UPDATED TRACKS RECOPIED.........2,639
FDR122                        DASD EXCPS......................381
FDR122                        TARGET DASD EXCPS...............450
FDR122                        CPU TIME (SECONDS)..............0.522
FDR122                        ELAPSED TIME (MINUTES)..........2.5
FDR122                        SWAP TIME.......................2.5
FDR999   FDR SUCCESSFULLY COMPLETED
```

- **DASD TRACKS SWAPPED** is the total number of tracks that had to be copied during the swap, including tracks that were copied more than once because of updates during the swap.
- **UPDATED TRACKS RECOPIED** is the number of tracks that had to be copied more than once because they were updated during the swap.

SWAP SEVERAL VOLUMES ON A SINGLE SYSTEM CONCURRENTLY EXAMPLE

Several volumes on adjacent z/OS addresses on a single system image are swapped to new devices on adjacent addresses (1AA*x*) in the new DASD subsystem. The volumes must not be accessible by any other system image. The target devices must be offline to all systems. All five volumes are swapped concurrently.

```
//SWAP     EXEC PGM=FDRPAS,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//FDRSUMM  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  SWAP     TYPE=FULL,MAXTASKS=5,CHECKTARGET=YES
  MOUNT    VOL=TSO001,SWAPUNIT=1AA*
  MOUNT    VOL=TSO002,SWAPUNIT=1AA*
  MOUNT    VOL=TSO003,SWAPUNIT=1AA*
  MOUNT    VOL=TSO004,SWAPUNIT=1AA*
  MOUNT    VOL=TSO005,SWAPUNIT=1AA*
/*
```

SIMULATE A SWAP AND GENERATE AND SUBMIT MONITOR TASKS EXAMPLE

Execute a simulated swap and generate and submit the MONITOR tasks for the other LPARs to validate the parameters you use for the real SWAP. The PASPROC procedure is used to ensure all the tasks use the same JCL and to reduce the number of JCL statements needed. The DURATION= operand is not coded since we want these MONITOR tasks to dynamically build the table to be used by the actual SWAP task that is submitted later.

RECOMMENDATION:

INNOVATION DATA PROCESSING recommends using the ISPF panels to generate the SIMSWAPMON job with the PASJOB DD statements.

If the PASJOB DD statement contains a `//*CPUID=` referring to the LPAR that the SWAP task is executing, that MONITOR task is not submitted to that LPAR since the SWAP task performs the monitoring. In this way, a single job can be created that can be submitted on any of the LPARS without requiring any change.

```
//PASSWAPM EXEC PASPROC
//PAS.SYSIN DD *
  SIMSWAPMON TYPE=FULL,MAXTASKS=16,LARGERSIZE=OK
  MOUNT    VOL=HI17CC,SWAPUNIT=17CA
  MOUNT    VOL=HI17C2,SWAPUNIT=17C1
/*
//PASJOB   DD DATA,DLM=$$
//*CPUID=0912342818
//PASMONA  JOB …
/*ROUTE    XEQ JESCPUA
//MONITORA EXEC PASPROC
//PAS.SYSIN DD *
  MONITOR  TYPE=SWAP,DYNMON=YES
/*
//*CPUID=0212342818
//PASMONB  JOB …
/*ROUTE    XEQ JESCPUB
//MONITORB EXEC PASPROC
//PAS.SYSIN DD *
  MONITOR  TYPE=SWAP,DYNMON=YES
/*
//*CPUID=0112342818
//PASMONC  JOB …
/*ROUTE    XEQ JESCPUC
//MONITORC EXEC PASPROC
//PAS.SYSIN DD *
  MONITOR  TYPE=SWAP,DYNMON=YES
/*
$$
```

SIMSWAPMON SAMPLE OUTPUT FROM PREVIOUS EXAMPLE

The following are examples of the output from the SWAP task and one of the MONITOR tasks. A MONITOR task was missing from one of the LPARs and is reflected in the output messages in the SWAP task.

The SYSPRINT output from the SWAP task looks similar to:

```
FDR001  FDR PLUG AND SWAP           - FDRPAS  VER. x.x/xx - INNOVATION DATA PROCESSING         DATE=yyyy.ddd PAGE    1
FDR303  CARD IMAGE --   SIMSWAPMON TYPE=FULL,MAXTASKS=16,LARGERSIZE=OK
FDR303  CARD IMAGE --   MOUNT    VOL=HI17CC,SWAPUNIT=17CA
FDR303  CARD IMAGE --   MOUNT    VOL=HI17C2,SWAPUNIT=17C1
FDR183  FDR SUBMITTED FDRPAS MONITOR JOBNAME=PASMONA  FOR CPUID=0912342818
FDR183  FDR SUBMITTED FDRPAS MONITOR JOBNAME=PASMONC  FOR CPUID=0112342818
FDR256  --- REPORT FROM SYSPRIN1 ---
FDR234** SWAP ERROR ON VOL=HI17CC - UNIT=17CA REASON=M - CPU (SERIAL#=0512342818) FAILED TO RESPOND IN PHASE 1
FDR234** SWAP ERROR ON VOL=HI17CC - UNIT=17CA REASON=Q - SWAP OPERATION CANCELLED NONRESPONDING=FAIL
FDR998** FDRPAS COMPLETED WITH ERRORS VOL=HI17CC
FDR256  --- REPORT FROM SYSPRIN2 ---
FDR234** SWAP ERROR ON VOL=HI17C2 - UNIT=17C1 REASON=M - CPU (SERIAL#=0512342818) FAILED TO RESPOND IN PHASE 1
FDR234** SWAP ERROR ON VOL=HI17C2 - UNIT=17C1 REASON=Q - SWAP OPERATION CANCELLED NONRESPONDING=FAIL
FDR998** FDRPAS COMPLETED WITH ERRORS VOL=HI17C2
FDR998** FDR COMPLETED WITH ERRORS
```

The SYSPRIN*n* DD statement output from the SWAP task looks similar to:

```
FDR001  FDR PLUG AND SWAP           - FDRPAS  VER. x.x/xx - INNOVATION DATA PROCESSING         DATE=yyyy.ddd PAGE    1
FDR234** SWAP ERROR ON VOL=HI17CC - UNIT=17CA REASON=M - CPU (SERIAL#=0512342818) FAILED TO RESPOND IN PHASE 1
FDR234** SWAP ERROR ON VOL=HI17CC - UNIT=17CA REASON=Q - SWAP OPERATION CANCELLED NONRESPONDING=FAIL
```

The SYSPRINT output from one of the MONITOR tasks looks similar to:

```
FDR001  FDR PLUG AND SWAP           - FDRPAS  VER. x.x/xx - INNOVATION DATA PROCESSING         DATE=yyyy.ddd PAGE    1
FDR303  CARD IMAGE --   MONITOR  TYPE=SWAP,DYNMON=YES
FDR235  FDRPAS ON CPU SERIAL# 0112342818 IS MONITORING DYNAMICALLY
FDR235  FDRPAS ON CPU SERIAL# 0112342818 IS MONITORING 17C1
FDR235  FDRPAS ON CPU SERIAL# 0112342818 IS MONITORING 17CA
FDR238   FDRPAS STARTED MONITOR JOIN TASK FOR UNIT=17C1
FDR238   FDRPAS STARTED MONITOR JOIN TASK FOR UNIT=17CA
FDR238   FDRPAS   ENDED MONITOR JOIN TASK FOR UNIT=17CA
FDR238   FDRPAS   ENDED MONITOR JOIN TASK FOR UNIT=17C1
FDR238   FDRPAS STARTED MONITOR JOIN TASK FOR UNIT=17C1
FDR238   FDRPAS STARTED MONITOR JOIN TASK FOR UNIT=17CA
FDR238   FDRPAS   ENDED MONITOR JOIN TASK FOR UNIT=17C1
FDR238   FDRPAS   ENDED MONITOR JOIN TASK FOR UNIT=17CA
```

The SYSPRT*nn* DD statement output from one of the MONITOR tasks looks similar to:

```
FDR001  FDR PLUG AND SWAP           - FDRPAS  VER. x.x/xx - INNOVATION DATA PROCESSING         DATE=yyyy.ddd PAGE    1
FDR303  CARD IMAGE -- MOUNT SU=17C1                                                                    ,,=17 PARM ENTRY
FDR235  FDRPAS ON CPU SERIAL# 0112342818 IS MONITORING THE FOLLOWING     1 UNITS:
FDR235   17C1
FDR233  CPUC     (SERIAL# 0112342818) ACKNOWLEDGES THE SWAP OF VOL=HI17C2 AND HAS JOINED IN SWAP OF UNIT=17C2 TO 17C1
FDR999   FDR SUCCESSFULLY COMPLETED

FDR001  FDR PLUG AND SWAP           - FDRPAS  VER. x.x/xx - INNOVATION DATA PROCESSING         DATE=2010.270 PAGE    1
FDR303  CARD IMAGE -- MOUNT SU=17C1                                                                    ,,=17 PARM ENTRY
FDR235  FDRPAS ON CPU SERIAL# 0112342818 IS MONITORING THE FOLLOWING     1 UNITS:
FDR235   17C1
FDR999   FDR SUCCESSFULLY COMPLETED
```

310.22 SWAPDUMP EXAMPLES

These are examples of SWAPDUMP operations that create a point-in-time image of an online volume. The point-in-time image can then be backed up with FDRINSTANT (or possibly by other backup software). All examples in this section are found in the JCL library installed with FDRPAS. The member names are PA31022*x*.

RECOMMENDATION: *INNOVATION DATA PROCESSING recommends using the GENSWAP procedure of processing a large number of volumes.*

POINT-IN-TIME BACKUP OF A VOLUME ON A SINGLE SYSTEM EXAMPLE

A volume accessible by a single system image is copied to an offline device. The volume must not be accessible by any other system image. As soon as FDRPAS finishes synchronizing the volumes, the target device becomes a frozen image of the source volume, at the point-in-time that FDRPAS finished. The second step uses FDRINSTANT to backup that point-in-time image; the special data set name of FDR.USE.UNIT07C3 on the DISK1 DD statement tells FDRINSTANT to backup the offline image on device 07C3.

```
//SWAPDUMP EXEC PGM=FDRPAS,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  SWAPDUMP TYPE=FULL
  MOUNT    VOL=DATA22,SWAPUNIT=07C3
/*
//BACKUP   EXEC PGM=FDR,REGION=0M,COND=(0,NE,SWAPDUMP)
//SYSPRINT DD SYSOUT=*
//DISK1    DD DSN=FDR.USE.UNIT07C3,UNIT=SYSALLDA,
//            VOL=SER=DATA22,DISP=OLD
//TAPE1    DD DSN=BACKUP.VDATA22,UNIT=TAPE,DISP=(,CATLG)
//SYSIN    DD *
  DUMP     TYPE=FDR
/*
```

POINT-IN-TIME BACKUP OF SEVERAL VOLUMES ON MULTIPLE SYSTEMS CONCURRENTLY EXAMPLE

Several volumes that are accessible to three system images are copied to offline devices. Prior to running these swaps, you must start an FDRPAS MONITOR task on each of the systems to monitor the target devices. The first example shows the use of the PASPROC cataloged procedure to start the SWAPDUMP operation by a START (S) command on the console. These run as system started tasks instead of jobs. Issue the commands shown on an appropriate console on the proper system. Note that CO= is an abbreviation for `CONFIRMSPLIT=` and `SU=` is an abbreviation for `SWAPUNIT=`, to reduce the length of the console input.

Because of the `CONFIRMSPLIT=YES`, FDRPAS waits for confirmation before freezing the point-in-time image. You can confirm the swap by using the FDRPAS ISPF panels, or by using a `MONITOR TYPE=CONFIRMSPLIT` job as shown in "Confirm the Swap of a Set of Volumes Example" in Section 310.24. Note that `CONFIRMSPLIT=YES` does not involve a WTOR to the console operator. Once the SWAPDUMP operations have been confirmed, you can use a product such as FDRINSTANT to backup those frozen images to tape.

```
S PASPROC.PROD01,PARM='SWAPDUMP TYPE=FULL,CO=YES/MOUNT VOL=PROD01,SU=1B32'
S PASPROC.PROD02,PARM='SWAPDUMP TYPE=FULL,CO=YES/MOUNT VOL=PROD02,SU=1B34'
S PASPROC.PROD03,PARM='SWAPDUMP TYPE=FULL,CO=YES/MOUNT VOL=PROD03,SU=1B37'
```

This is an alternate way of accomplishing the same SWAPDUMPs, by submitting a single batch job using PASPROC. All three volumes are copied concurrently by this one job, since `MAXTASKS=` is specified. You can add additional MOUNT statements to do more volumes concurrently.

```
//SWAP     EXEC PASPROC
//PAS.SYSIN DD *
  SWAPDUMP TYPE=FULL,MAXTASKS=20,CONFIRMSPLIT=YES
  MOUNT    VOL=PROD01,SWAPUNIT=1B32
  MOUNT    VOL=PROD02,SWAPUNIT=1B34
  MOUNT    VOL=PROD03,SWAPUNIT=1B37
/*
```

Here is another alternate way of accomplishing the same swaps, using a started task. If the SWAPDUMP and MOUNT statements in the example above are stored in a member of a PDS, you can use:

```
S PASPROC.PRODVOLS,IN=SYSPROG.FDRPAS(PRODVOLS)
```

NOTE: Contact INNOVATION DATA PROCESSING for the *FDRPAS SWAPDUMP for Duplicating a SYSTEM* document if processing a large number of volumes.

310.23 MONITOR SWAP EXAMPLES

If the source volume is accessible to multiple systems, even if it is offline on some of those systems, an FDRPAS MONITOR task, as shown in the following examples, must be started on every one of those systems before an FDRPAS SWAP, SWAPDUMP, SIMSWAPMON, SWAPBUILDIX, or EXPANDVTOC task is started. If the source volume is in an EMC Symmetrix subsystem, the MONITOR tasks need to be started only on systems where the source volume is online. On other DASD subsystem types, the MONITOR tasks must execute on all systems with access to the source volume, even if it is offline.

The MONITOR tasks shown below are used with SWAP, SWAPDUMP, SIMSWAPMON, SWAPBUILDIX, and EXPANDVTOC operations. Although the text below refers to "SWAP tasks" the examples work with all these functions.

These MONITOR tasks can be started automatically by the main SWAP, SWAPDUMP, or SIMSWAPMON job if in GRS complexes or MIM complexes.

All examples in this section are found in the JCL library installed with FDRPAS. The member names are PA31023*x*.

DYNAMIC MONITORING EXAMPLE

This FDRPAS MONITOR task dynamically monitors the volumes involved with a SWAP, SWAPDUMP, or SIMSWAPMON. This method only takes resources when a main swap process specifies a target device for a swap. Since a dynamic monitor only uses resources when swapping, it could be started at IPL time and remain available.

```
//MONITOR  EXEC PGM=FDRPAS,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//FDRSUMM  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  MONITOR  TYPE=SWAP,DYNMON=YES
/*
```

NOTE: You should specify `DYNMON=YES` or monitor multiple units if you are going to execute SIMSWAPMON.

MONITOR A SINGLE TARGET DEVICE EXAMPLE

Monitor offline device 1AA3 as an FDRPAS target device. FDRPAS periodically checks this device to see if an FDRPAS SWAP task has started on another z/OS system image that is using the device as a target. If so, the MONITOR task assists the SWAP task by monitoring this system image for updates to the original volume during the copy process, and by switching all I/O activity to the new device when the swap is complete.

This type of MONITOR task automatically terminates once the SWAP or SIMSWAPMON job has completed processing the target device.

```
//MONITOR  EXEC PASPROC
//PAS.SYSIN DD *
  MONITOR  TYPE=SWAP
  MOUNT    SWAPUNIT=1AA3
/*
```

MONITOR A RANGE OF TARGET DEVICES EXAMPLE #1

Monitor range 24E0-24EF of offline potential FDRPAS target devices (any devices in the range that are not offline or are not DASD are ignored). FDRPAS periodically checks each of those devices to see if an FDRPAS SWAP task has started on another system image that is using the device as a target. If so, it starts a separate MONITOR task to assist the SWAP task by monitoring this system image for updates to the original volume during the copy process, and by switching all I/O activity to the new device when the swap is complete. Once all swaps on this range of offline devices have been completed, you can terminate the MONITOR task with console command `P MON2`. However, if every offline device in the range is now online (completed swap) the MONITOR task terminates automatically.

```
S PASPROC.MON2,PARM='MONITOR TYPE=SWAP/MOUNT SU=24E*'
```

If your systems are part of a sysplex (basic or parallel), you can use the console ROUTE command to start MONITOR tasks on all systems in the sysplex. This can considerably reduce the amount of typing required. For example:

```
ROUTE T=0,*OTHER,S PASPROC.MON2,PARM='MONITOR TYPE=SWAP/MOUNT SU=24E*'
```

starts the MONITOR task on all other systems in the sysplex (*OTHER assumes that you start the SWAP task on this system, use *ALL to start the MONITOR task on all systems).

If you have systems that are not part of the sysplex, you need to manually start the MONITOR task on each of them.

MONITOR A RANGE OF TARGET DEVICES EXAMPLE #2

Monitor ranges (1A00-1AFF and 1B00-1BFF) of offline potential FDRPAS target devices (any devices in the range that are not offline or not DASD are ignored). FDRPAS periodically checks each of those devices to see if an FDRPAS SWAP task has started on another system image that is using the device as a target. If so, it starts a separate MONITOR task to assist the SWAP task by monitoring this system image for updates to the original volume during the copy process, and by switching all I/O activity to the new device when the swap is complete. As long as no more than 64 concurrent swaps are occurring on the monitored devices, all the separate MONITOR tasks are internal subtasks. The MONITOR task automatically terminates when it has been idle (not participating in any swaps) for a total of two hours (120 minutes).

NOTE: This MONITOR task needs to be started on every system that has access to the volumes to be swapped before the SWAP task is started.

```
//MONITOR  EXEC PGM=FDRPAS,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  MONITOR  TYPE=SWAP,DURATION=120
  MOUNT    SWAPUNIT=(1A*,1B*)
/*
```

310.24 MONITOR CONFIRM EXAMPLES

All examples in this section are found in the JCL library installed with FDRPAS. The member names are PA31024*x*.

RECOMMENDATION: *That the ISPF panels be used to generate the FDRPAS MONITOR CONFIRM JCL and control statements.*

CONFIRM THE SWAP OF A SET OF VOLUMES EXAMPLE

FDRPAS SWAP tasks are currently swapping volumes TSO001 to TSO005 and they specified `CONFIRMSWAP=YES`, so they wait for a confirmation after the volumes are synchronized. The following job causes an automatic confirmation of the swap. If it is submitted before the volumes are synchronized, it waits for them to all become synchronized, and then it completes the swap of all five volumes, at one time. If the volumes are already synchronized when it is submitted, it completes the swaps immediately.

NOTE: `CONFIRMSWAP=YES` is not recommended unless you have some real need to cause the swaps to complete all at the same time. Usually, it is better to let each swap complete automatically as soon as the devices are synchronized.

```
//CONFIRM  EXEC PGM=FDRPAS,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  MONITOR  TYPE=CONFIRMSWAP
  MOUNT    VOL=TSO001
  MOUNT    VOL=TSO002
  MOUNT    VOL=TSO003
  MOUNT    VOL=TSO004
  MOUNT    VOL=TSO005
/*
```

CONFIRM THE SWAPDUMP OF A SET OF VOLUMES EXAMPLE

FDRPAS SWAP tasks are currently creating point-in-time backups (SWAPDUMP) of volumes PROD01, PROD02, and PROD03 and they specified `CONFIRMSPLIT=YES`, so they wait for a confirmation after the volumes are synchronized (see "Confirm SWAPDUMP with Operator Confirmation Example" in Section 310.24). The following job causes an automatic confirmation of the operation. If it is submitted before the volumes are synchronized, it waits for them to all become synchronized, and then it completes the creation of the point-in-time backups of all three volumes, at one time. If the volumes are already synchronized when it is submitted, it completes the point-in-time backups immediately. The following step executes FDRINSTANT to create tape backups of the point-in-time backups; the special data set name of FDR.USE.UNIT*uuuu* on the DISK*x* DD statements tells FDRINSTANT to backup the offline image on device "*uuuu*".

```
//CONFIRM  EXEC PGM=FDRPAS,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  MONITOR  TYPE=CONFIRMSPLIT
  MOUNT    VOL=PROD01
  MOUNT    VOL=PROD02
  MOUNT    VOL=PROD03
/*
//BACKUP   EXEC PGM=FDR,REGION=0M,COND=(0,NE,CONFIRM)
//SYSPRINT DD SYSOUT=*
//SYSPRIN1 DD SYSOUT=*
//SYSPRIN2 DD SYSOUT=*
//SYSPRIN3 DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//DISK1    DD DSN=FDR.USE.UNIT1B32,UNIT=SYSALLDA,
//            VOL=SER=PROD01,DISP=OLD
//TAPE1    DD DSN=BACKUP.VPROD01,UNIT=TAPE,DISP=(,CATLG)
//DISK2    DD DSN=FDR.USE.UNIT1B34,UNIT=SYSALLDA,
//            VOL=SER=PROD02,DISP=OLD
//TAPE2    DD DSN=BACKUP.VPROD02,UNIT=TAPE,DISP=(,CATLG)
//DISK3    DD DSN=FDR.USE.UNIT1B37,UNIT=SYSALLDA,
//            VOL=SER=PROD03,DISP=OLD
//TAPE3    DD DSN=BACKUP.VPROD03,UNIT=TAPE,DISP=(,CATLG)
//SYSIN    DD *
  DUMP     TYPE=FDR,ATTACH
/*
```

CONFIRM SWAPDUMP WITH OPERATOR CONFIRMATION EXAMPLE

FDRPAS SWAP tasks are currently creating point-in-time backups (SWAPDUMP) of a large set of volume, with `CONFIRMSPLIT=YES` specified. Once all of the SWAPDUMP tasks have been submitted and are executing, you can submit the CONFIRM step below. Because `CONFMESS=YES` is specified, the CONFIRM step does not immediately confirm the SWAPDUMPs. Rather, it issues WTOR message FDRW07 to the system operator to indicate when the volumes are ready to confirm, meaning that all the volumes are synchronized. When the operator replies "YES" to the WTOR, the volumes are confirmed and the SWAPDUMPs are completed.

This can be useful when a system automation product is used. The automation product can recognize the FDRW07 message, take other actions such as quiescing applications, and then reply to the message to confirm the SWAPDUMPs.

```
//CONFIRM  EXEC PGM=FDRPAS,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  MONITOR  TYPE=CONFIRMSPLIT,MAXCARDS=500,CONFMESS=YES
  MOUNT    VOL=ABC001
  MOUNT    VOL=XYZ987
/*
```

310.25 HISTORY EXAMPLES

All examples in this section are found in the JCL library installed with FDRPAS. The member names are PA31025*x*.

DISPLAY HISTORY FOR A SET OF VOLUMES EXAMPLE

FDRPAS displays swap history records for all volumes beginning with PROD or DB.

```
//HISTORY  EXEC PGM=FDRPAS,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  HISTORY  TYPE=SWAP
  MOUNT    VOL=PROD*
  MOUNT    VOL=DB*
/*
```

DISPLAY AND PURGE HISTORY FOR ALL VOLUMES EXAMPLE

FDRPAS displays swap history records for all volumes that FDRPAS has swapped. It also purges history records that are older than 30 days.

```
//HISTORY  EXEC PGM=FDRPAS,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  HISTORY  TYPE=SWAP,MAXSWAPHISTORYDAYS=30
  MOUNT    VOL=*
/*
```

310.26 SWAPBUILDIX AND EXPANDVTOC EXAMPLES

All examples in this section are found in the JCL library installed with FDRPAS. The member names are PA31026*x*.

ADD A VTOCIX TO A VOLUME EXAMPLE

An Indexed VTOC is allocated and built on a volume that does not currently have a VTOCIX. FDRPAS determines the number of systems that have access to the volume. Prior to running this job, you must start FDRPAS MONITOR tasks on each of the systems to monitor the offline monitor device specified by SWAPUNIT= (see example below). The monitor device must be offline on all systems.

```
//BUILDIX  EXEC PGM=FDRPAS,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//VTOCIX   DD DSN=SYS1.VTOCIX.DATA22,DISP=(,KEEP),UNIT=SYSALLDA,
//            VOL=SER=DATA22,SPACE=(TRK,5)      (see note)
//SYSIN    DD *
  SWAPBUILDIX TYPE=FULL
  MOUNT    VOL=DATA22,SWAPUNIT=07C3
/*
```

NOTE: Consult the *IBM Device Support Facilities (ICKDSF) User's Guide and Reference (GC35-0033)* for information on allocating and sizing VTOCIX data sets.

REBUILD THE VTOCIX ON THREE VOLUMES EXAMPLE

The disabled Indexed VTOC is rebuilt on three volumes. FDRPAS determines the number of systems that have access to each volume. Prior to running this job, you must start FDRPAS MONITOR tasks on each of the systems to monitor the offline monitor device specified by SWAPUNIT= (see example below). The monitor device must be offline on all systems.

```
//BUILDIX  EXEC PGM=FDRPAS,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  SWAPBUILDIX TYPE=FULL
  MOUNT    VOL=DATA23,SWAPUNIT=07C3
  MOUNT    VOL=DATA24,SWAPUNIT=07C3
  MOUNT    VOL=DATA25,SWAPUNIT=07C3
/*
```

EXPAND THE VTOC ON TWO VOLUMES EXAMPLE

The VTOC is expanded to a new size on two volumes. The VTOCIX is sized to 1/16th of the size of the VTOC unless the VTOCIX is already larger. FDRPAS determines the number of systems that have access to each volume. Prior to running this job, you must start FDRPAS MONITOR tasks on each of the systems to monitor the offline monitor device specified by SWAPUNIT= (see example below). The monitor device must be offline on all systems.

```
//EXPVTOC  EXEC PGM=FDRPAS,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  EXPANDVTOC  TYPE=FULL
  MOUNT    VOL=TSO002,SWAPUNIT=07C3,VTOCSIZE=650
  MOUNT    VOL=TSO003,SWAPUNIT=07C3,VTOCSIZE=650
/*
```

NOTE: Within one job you can use the same SWAPUNIT= address on every MOUNT statement; the SWAPUNIT= address is used only for communication between the LPARs. However, if you submit multiple EXPANDVTOC jobs, you must specify a different SWAPUNIT= in each one. INNOVATION DATA PROCESSING does not recommend that you run multiple EXPANDVTOC jobs. You must be licensed for FDRMOVE for this feature.

MONITOR THE SWAPBUILDIX OR EXPANDVTOC MONITOR DEVICE EXAMPLE

Monitor offline device 07C3 for SWAPBUILDIX or EXPANDVTOC operations; this is actually the same as MONITOR tasks used with other FDRPAS operations. FDRPAS periodically checks this device to see if an FDRPAS SWAPBUILDIX or EXPANDVTOC task has started on another system. If so, the MONITOR task waits for the VTOCIX to be rebuilt and then update the VTOCIX information on this system.

A MONITOR task that monitors only one SWAPUNIT continues to look for SWAPBUILDIX or EXPANDVTOC operations until one minute goes by with no requests, and then terminates automatically. If you want the MONITOR to keep monitoring for a longer interval, specify a sufficient value for the `DURATION=` parameter, for example,

```
MONITOR TYPE=SWAP,DURATION=1000
```

then when you are done, terminate the MONITOR with the console command "`P` *`jobname`*".

```
//MONITOR  EXEC PASPROC
//PAS.SYSIN DD *
  MONITOR  TYPE=SWAP
  MOUNT    SWAPUNIT=07C3
/*
```

310.27 VARY EXAMPLES

All examples in this section are found in the JCL library installed with FDRPAS. The member names are PA31027*x*.

RELABEL AND MOUNT SWAP SOURCE VOLUMES EXAMPLE

Three volumes are swapped to new devices, leaving the old DASD devices as point-in-time backups of the volumes at the time of the swap completion but there is a need to access those backups. This job relabels the source volumes with new volume serials, fixes the volume label so they can be mounted, and varies them online.

```
//VARYON   EXEC PGM=FDRPAS,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  MONITOR  TYPE=VARYONLINE
  MOUNT    VARYUNIT=07C3,NVOL=OLD7C3
  MOUNT    VARYUNIT=07C4,NVOL=OLD7C4
  MOUNT    VARYUNIT=07C5,NVOL=OLD7C5
/*
```

RELABEL AND MOUNT SWAPDUMP TARGET VOLUMES EXAMPLE

SWAPDUMP was successfully executed against sixteen DASD volumes, creating offline point in time copies. These copies can be backed up with FDRINSTANT, but if you need to execute other utilities against the copies, this job relabels the copied volumes with new volume serials, fixes the volume label so they can be mounted, and varies them online. The original volumes had volsers of DB2*xxx*, so the new volumes have serials of DBC*xxx*.

```
//VARYON   EXEC PGM=FDRPAS,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  MONITOR  TYPE=VARYONLINE
  MOUNT    VARYUNIT=21C*,NVOL=**C***
/*
```

RELABEL OFFLINE VOLUME EXAMPLE

This job re-labels an offline volume with a new volume serial and fixes the volume label so it can be mounted, but does not vary it online. It can be varied online to another system, if desired. This can be executed against a SWAP source volume or a SWAPDUMP target device. If you change the `NVOL` operand to `NVOL=******`, FDRPAS just changes the value of FDR1 in the label track of the volume to VOL1 so that the volume can be varied online.

```
//RESETVOL EXEC PGM=FDRPAS,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  MONITOR  TYPE=RESETVOL
  MOUNT    VARYUNIT=7C12,NVOL=TMPVOL
/*
```

310.30 FDRPAS SAMPLE PRINTOUTS

SAMPLE SWAP TASK PRINTOUT

In this sample printout below from an FDRPAS SWAP task, you can see:

- The control statements, requesting the swap.
- MONITOR tasks on ten additional systems joining in the swap (Phase 1).
- I/O intercepts installed on all 11 systems (Phase 2).
- Active data sets are identified (note that since SYSDSN enqueues are not specific by volume serial number, if you are swapping a volume containing uncataloged copies of data sets that are active on another volume, FDRPAS may identify them as active on this volume).
- The initial copy of active data tracks (Phase 3, pass 1).
- Final copying of updated tracks (Phase 4 and Phase 3, pass 2).
- Swap completed (Phase 5).
- Statistics about the swap.

```
FDR001  FDR PLUG AND SWAP     - FDRPAS                - INNOVATION DATA PROCESSING                      PAGE   1
FDR303  CARD IMAGE --    SWAP TYPE=FULL
FDR303  CARD IMAGE --    MOUNT VOL=TSO010,SWAPUNIT=226F

FDR233  SYSA  (SERIAL# 0109419672) ACKNOWLEDGES THE SWAP OF VOL=TSO010
FDR233  SYSC  (SERIAL# 0132429672) ACKNOWLEDGES THE SWAP OF VOL=TSO010 AND HAS JOINED IN SWAP OF UNIT=2121 TO 226F
FDR233  SYSJ  (SERIAL# 0209419672) ACKNOWLEDGES THE SWAP OF VOL=TSO010 AND HAS JOINED IN SWAP OF UNIT=2121 TO 226F
FDR233  SYSZ  (SERIAL# 0054502064) ACKNOWLEDGES THE SWAP OF VOL=TSO010 AND HAS JOINED IN SWAP OF UNIT=2121 TO 226F
FDR233  SYSE  (SERIAL# 0309419672) ACKNOWLEDGES THE SWAP OF VOL=TSO010 AND HAS JOINED IN SWAP OF UNIT=2121 TO 226F
FDR233  SYSD  (SERIAL# 0146279672) ACKNOWLEDGES THE SWAP OF VOL=TSO010 AMD HAS JOINED IN SWAP OF UNIT=2121 TO 226F
FDR233  SYSH  (SERIAL# 0270039672) ACKNOWLEDGES THE SWAP OF VOL=TSO010 AND HAS JOINED IN SWAP OF UNIT=2121 TO 226F
FDR233  SYSI  (SERIAL# 0032429672) ACKNOWLEDGES THE SWAP OF VOL=TSO010 AND HAS JOINED IN SWAP OF UNIT=2121 TO 226F
FDR233  SYSB  (SERIAL# 0145399672) ACKNOWLEDGES THE SWAP OF VOL=TSO010 AND HAS JOINED IN SWAP OF UNIT=2121 TO 226F
FDR233  SYSK  (SERIAL# 0432429672) ACKNOWLEDGES THE SWAP OF VOL=TSO010 AND HAS JOINED IN SWAP OF UNIT=2121 TO 226F
FDRW66  SWAP OF VOL=TSO010 TO UNIT=226F STARTED ON  10 SYSTEMS (SYSA SYSC SYSJ SYSZ SYSE SYSD SYSH SYSI SYSB SYSK)

FDR236  SYSA  ACTIVATED I/O INTERCEPTS ON UNIT=2121
FDR007  STARTING TIME OF FULL VOL SWAP -- 10.17.59 -- UNIT=3390  ,IN=D#TSO010,OUTPUT=TAPE1
FDR158  DATA SET IS ACTIVE  DSN=TSO.USER.TEST
FDR239      106 TRACKS UPDATED BY SYSA
FDR239        8 TRACKS UPDATED BY SYSJ
FDR239        7 TRACKS UPDATED BY SYSH
FDR239      109 TOTAL UNIQUE TRACKS UPDATED IN PASS     1 - RE-COPYING UPDATED TRACKS
FDR236  SYSA  DE-ACTIVATED I/O INTERCEPTS ON UNIT=2121       106 TRACKS UPDATED

FDR241  FDRPAS SUCCESSFULLY COMPLETED SWAP OF VOL=TSO010 TO UNIT=226F ON SYSA
FDR241  FDRPAS SUCCESSFULLY COMPLETED SWAP OF VOL=TSO010 TO UNIT=226F ON SYSC
FDR241  FDRPAS SUCCESSFULLY COMPLETED SWAP OF VOL=TSO010 TO UNIT=226F ON SYSJ
FDR241  FDRPAS SUCCESSFULLY COMPLETED SWAP OF VOL=TSO010 TO UNIT=226F ON SYSZ
FDR241  FDRPAS SUCCESSFULLY COMPLETED SWAP OF VOL=TSO010 TO UNIT=226F ON SYSE
FDR241  FDRPAS SUCCESSFULLY COMPLETED SWAP OF VOL=TSO010 TO UNIT=226F ON SYSD
FDR241  FDRPAS SUCCESSFULLY COMPLETED SWAP OF VOL=TSO010 TO UNIT=226F ON SYSH
FDR241  FDRPAS SUCCESSFULLY COMPLETED SWAP OF VOL=TSO010 TO UNIT=226F ON SYSI
FDR241  FDRPAS SUCCESSFULLY COMPLETED SWAP OF VOL=TSO010 TO UNIT=226F ON SYSB
FDR241  FDRPAS SUCCESSFULLY COMPLETED SWAP OF VOL=TSO010 TO UNIT=226F ON SYSK
FDR007  ENDING   TIME OF FULL VOL SWAP -- 10.24.24 -- UNIT=3390 ,IN=D#TSO010,OUTPUT=TAPE1
FDR122  OPERATION STATISTICS FOR 3390 VOLUME.................TSO010
FDR122                        CYLINDERS ON VOLUME.............3,339
FDR122                        DATASETS PROCESSED..............274
FDR122                        BYTES READ FROM DASD.....1,467,108,993
FDR122                        DASD TRACKS SWAPPED.............29,098
FDR122                        UPDATED TRACKS RECOPIED............109
FDR122                        DASD EXCPS.....................1,991
FDR122                        TARGET DASD EXCPS..............2,080
FDR122                        CPU TIME (SECONDS)..............2.297
FDR122                        ELAPSED TIME (MINUTES).............6.5
FDR122                        SWAP TIME.........................6.3
FDR999   FDR SUCCESSFULLY COMPLETED
```

SAMPLE SWAP SUMMARY OUTPUT

The summary output appears in a SWAP task if an FDRSUMM DD statement is present. It contains a one-line summary for each volume processed, showing the completion code, elapsed time, volume size in cylinders, bytes read, data sets on the volume at the start of the swap, tracks updated during the swap and total tracks copied.

```
FDR001  FDR PLUG AND SWAP           - FDRPAS                - INNOVATION DATA PROCESSING
        COMP ELAPSED          VOLUME     DASD BYTES       DATASETS UPDATED     TRACKS
VOLSER  CODE TIME(MIN)          SIZE   READ FROM VOL      PROCESSED TRACKS     SWAPPED
SH20CC     0      7.2          1,113     662,278,052            61 10724       14,173
SH20CE     0     12.3          3,339   1,833,760,322           355   314       27,022
```

SAMPLE SWAP MONITOR OUTPUT

In this sample printout below from a user-submitted FDRPAS MONITOR task, you can see:

- The control statements requesting that FDRPAS monitor a range of target device, waiting for swaps to begin.
- The MONITOR task detected swaps beginning on four target devices, at different times.
- Three swaps completed successfully on this system.
- The MONITOR task terminated automatically after two hours (`DURATION=120`).

```
FDR303  CARD IMAGE --  MONITOR TYPE=SWAP,DURATION=120
FDR303  CARD IMAGE --  MOUNT SWAPUNIT=21C*
FDR235  FDRPAS ON CPU SERIAL# 0432429672 IS MONITORING THE FOLLOWING    16 UNITS:
FDR235   21C0 21C1 21C2 21C3 21C4 21C5 21C6 21C7 21C8 21C9 21CA 21CB 21CC 21CD 21CE 21CF

FDR238   FDRPAS STARTED MONITOR JOIN TASK FOR UNIT=21C2
FDR241  FDRPAS SUCCESSFULLY COMPLETED SWAP OF VOL=RVA7D2 TO UNIT=21C2 ON CPUB

FDR238   FDRPAS STARTED MONITOR JOIN TASK FOR UNIT=21C1
FDR241  FDRPAS SUCCESSFULLY COMPLETED SWAP OF VOL=RVA7D1 TO UNIT=21C1 ON CPUB

FDR238   FDRPAS STARTED MONITOR JOIN TASK FOR UNIT=21C7
FDR241  FDRPAS SUCCESSFULLY COMPLETED SWAP OF VOL=RVA7D7 TO UNIT=21C7 ON CPUB
FDR999   FDR SUCCESSFULLY COMPLETED
```

In this sample printout below from an internally-started FDRPAS MONITOR task, you can see:

- The control statements, requesting that FDRPAS join a swap that was detected by the user-submitted MONITOR task.
- The swap beginning on another system and this system joining in the swap (Phase 1).
- I/O intercept installed on this system (Phase 2).
- I/O intercept de-installed on this system (Phase 4). While active, it detected that 252 tracks on the source volume were updated on this system.
- Swap completed (Phase 5).

The output from a user-submitted MONITOR task that monitors only a single target device is similar to this example.

```
FDR001  FDR PLUG AND SWAP           - FDRPAS                - INNOVATION DATA PROCESSING                    PAGE    1
FDR303  CARD IMAGE -- MONITOR TYPE=JOIN,NOS/MOUNT SU=21C1                                                  PARM ENTRY
FDR303  CARD IMAGE -- MOUNT SU=21C1                                                                  ,,    PARM ENTRY
FDR235  FDRPAS ON CPU SERIAL# 0432429672 IS MONITORING THE FOLLOWING    1 UNITS:
FDR235   21C1
FDR233  CPUB      (SERIAL# 0209417060) ACKNOWLEDGES THE SWAP OF VOL=RVA7D1 AND HAS JOINED IN SWAP OF UNIT=07D1 TO 21C1
FDR236  CPUB      ACTIVATED I/O INTERCEPTS ON UNIT=07D1
FDR236  CPUB      DE-ACTIVATED I/O INTERCEPTS ON UNIT=07D1       252 TRACKS UPDATED
FDR241  FDRPAS SUCCESSFULLY COMPLETED SWAP OF VOL=RVA7D1 TO UNIT=21C1 ON CPUB
FDR999   FDR SUCCESSFULLY COMPLETED
```

310.31 FDRPAS ISPF INTERFACE

The FDRPAS ISPF interface allows you to initiate, monitor, and control FDRPAS operations on the system to which your TSO session is logged on. You can:

- Monitor active swaps
- Initiate SWAP, SWAPDUMP, MONITOR, SIMSWAP, and SIMSWAPMON tasks
- Confirm swaps that specified CONFIRMSWAP=YES or CONFIRMSPLIT=YES
- Suspend and resume active swaps
- Terminate active swaps
- Reply to certain FDRPAS messages
- Display FDRPAS history records
- Display basic information about any DASD devices in your installation, selecting them by unit address, volume serial number, subsystem ID, subsystem serial number, or SMS storage group.

INVOKING THE FDRPAS ISPF DIALOGS

The FDRPAS ISPF dialogs are integrated with the ISPF dialogs of FDR, the INNOVATION DATA PROCESSING FDR DASD management system. If you have installed the FDR ISPF dialogs at a release level matching the level of FDRPAS that you are using, your FDR or ABR main ISPF menu has an option “P” for FDRPAS. If so, you can skip the invocation of ABRALLOC shown below and use that option to invoke the FDRPAS dialogs.

If you do not have an appropriate level of the FDR dialogs installed, or they are at a lower release level than the level of FDRPAS in use; then use the following procedure:

Go to ISPF option 6, or exit ISPF to TSO READY mode, and issue this command:

```
EXEC 'fdrpas.clist.library(ABRALLOC)'
```

specifying the name of the FDRPAS CLIST library that was installed during the installation of FDRPAS (see Section 380.2 “Electronic Installation” through Section 380.3 “”). This allocates all of the required FDRPAS ISPF libraries and invoke the ABR dialogs (including FDRPAS).

If you issued the ABRALLOC from TSO READY, you see a standard ISPF main menu with an additional option “A” (for ABR). If your installation has modified your ISPF main menu, this may look considerably different than you are normally used to seeing. This ISPF menu looks something like this:

ISPF PRIMARY OPTION MENU

```
  Menu  Utilities  Compilers  Options  Status  Help
------------------------------------------------------------------------------
                          ISPF Primary Option Menu
Option ===> A

0  Settings      Terminal and user parameters          User ID . : IDP
1  View          Display source data or listings       Time. . . : 09:05
2  Edit          Create or change source data          Terminal. : 3278
3  Utilities     Perform utility functions             Screen. . : 1
4  Foreground    Interactive language processing       Language. : ENGLISH
5  Batch         Submit job for language processing    Appl ID . : ISR
6  Command       Enter TSO or Workstation commands     TSO logon : V59ISPF
7  Dialog Test   Perform dialog testing                TSO prefix: IDP
8  LM Facility   Library administrator functions       System ID : CPUB
9  IBM Products  IBM program development products      MVS acct. : **NONE**
10 SCLM          SW Configuration Library Manager      Release . : ISPF 5.9
11 Workplace     ISPF Object/Action Workplace
A  FDR/ABR       FDR/ABR DASD Management Functions
```

Select option “A” on the Option line, as shown and press ENTER to display the FDR/ABR primary options menu.

FDR PRIMARY OPTIONS MENU

The FDR primary options menu looks similar to this:

FDR PRIMARY OPTIONS MENU – PANEL A

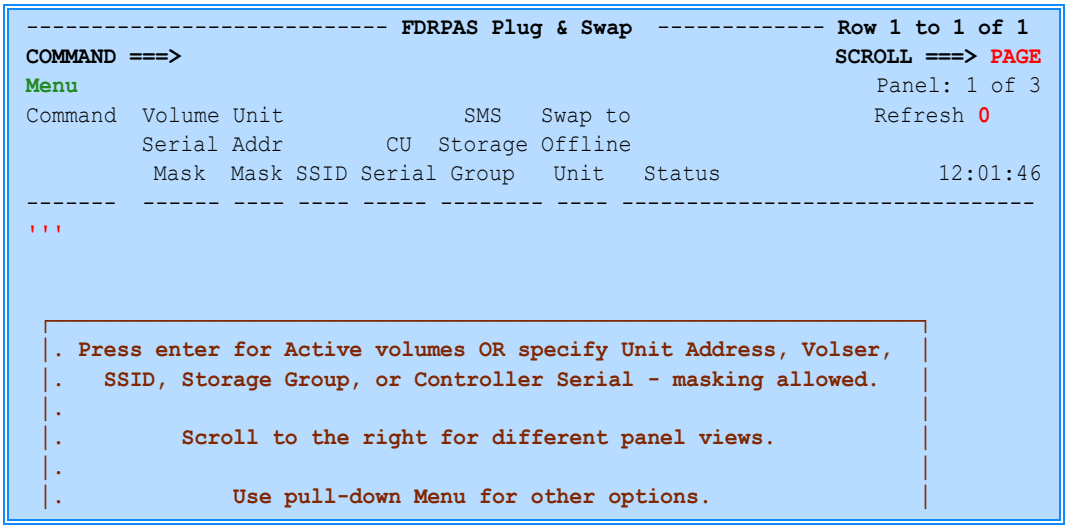
```
--------- FDR TOTAL DASD MANAGEMENT SYSTEM -- FDR PRIMARY OPTIONS MENU ------
 OPTION  ===>
                                                                      V 5.4/80

  1  REPORTS     - ABR REPORTING FUNCTIONS
  2  RESTORE     - ABR DATA SET RESTORE
  3  ARCHIVE     - ABR DATA SET ARCHIVE OR SUPERSCRATCH
  4  BACKUP      - ABR DATA SET BACKUP
  5  REMOTE Q    - ABR REMOTE QUEUE UTILITY FUNCTIONS

  C  COMPAKTOR   - COMPAKTOR MAP AND SIMULATION REPORTS
  R  RELEASE     - COMPAKTOR RELEASE

  I  INSTALL     - INSTALLATION AND MAINTENANCE OF FDR AND OPTIONAL PRODUCTS
  J  JCL PARMS   - SPECIFY FDR JCL AND SYSOUT DEFAULTS FOR SUBMITTED JOBS
  K  FORMAT      - MODIFY FORMAT OF GENERATED REPORTS

  P  PLUG & SWAP - FDRPAS PLUG & SWAP
  E  FDRERASE    - FDR DISK ERASE
  M  FDRMOVE     - FDRMOVE PLUG & SWAP DATA SET MOVE

  MC MESSAGES    - FDR MESSAGES AND CODES QUERY FACILITY
  Q  QUERY       - FDR/ABR STATISTICS QUERY
  S  SRS         - SEARCH, REPORT, SERVICES DIALOG
  T  FDRTSEL     - BACKUP FILE MANAGEMENT UTILITY
```

Select option “P” to monitor and control FDRPAS. Other options are used for FDRABR and other components of the FDR DASD management software. Option “I” is used during installation of FDRPAS (see Chapter 380 “Installation & Maintenance”).

FDRPAS PANEL

This panel is displayed by the "P" option. It is used to initiate, monitor, and control FDRPAS operations on the system to which your TSO session is logged on.

FDRPAS PLUG & SWAP – PANEL A.P

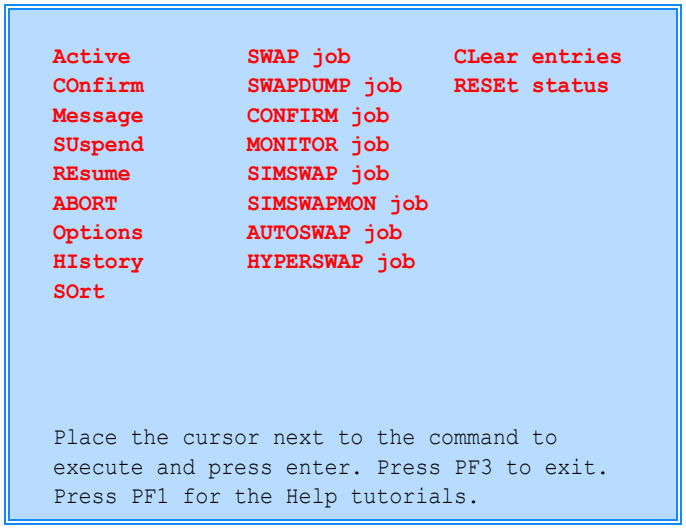

FDRPAS Plug & Swap
Row 1 to 1 of 1
COMMAND ===>
SCROLL ===> PAGE
Menu
Panel: 1 of 3
Command Volume Unit SMS Swap to
Refresh 0
Serial Addr CU Storage Offline
Mask Mask SSID Serial Group Unit Status
12:01:46
'''
. Press enter for Active volumes OR specify Unit Address, Volser,
. SSID, Storage Group, or Controller Serial - masking allowed.
. Scroll to the right for different panel views.
. Use pull-down Menu for other options.

You can:

- Monitor active swaps.
- Generate JCL streams for operations such as SWAP, SWAPDUMP, CONFIRM, MONITOR, SIMSWAP, SIMSWAPMON, AutoSwap, and HyperSwap.
- Confirm swaps where `CONFIRMSWAP=YES` or `CONFIRMSPLIT=YES` is specified.
- Suspend and resume swaps.
- Terminate active swaps.
- Reply to certain FDRPAS console messages.
- Display FDRPAS history records.
- Display basic information about any DASD devices in your installation, selecting by unit address, volume serial number, subsystem ID, subsystem serial number, or SMS storage group.

Positioning the cursor on "Menu" and pressing ENTER displays a "pull-down" list of available commands that can be selected or typed on the command line or the command area of each row.

FDRPAS PLUG & SWAP – COMMAND PULL-DOWN MENU



In the "pull-down" menu, the first column are the commands:

Active	Display volumes currently being processed by FDRPAS.
COnfirm	Respond to confirm volumes waiting for `CONFIRMSWAP` or `CONFIRMDUMP`.
Message	Display a panel where confirmation messages can be replied to.
SUspend	Suspend the copy between the source to the target volume.
REsume	Resume the copy between the source and the target volume.
ABORT	Immediately terminate an active swap.
Options	Display the Options panel.
HIstory	Get history for previous swaps.
SOrt	Display a panel where sort criteria for fields are specified.

The second column is used to generate batch jobs:

SWAP job	Generate a SWAP JCL stream.
SWAPDUMP job	Generate a SWAPDUMP JCL stream.
CONFIRM job	Generate a CONFIRM JCL stream.
MONITOR job	Generate a MONITOR JCL stream.
SIMSWAP job	Generate a SIMSWAP JCL stream.
SIMSWAPMON job	Generate a SIMSWAPMON JCL stream.
AutoSwap job	Generate an AutoSwap JCL stream.
HYPERSWAP job	Generate a HyperSwap JCL stream.

The third column has the commands:

CLear	Clear all volume entries in the display. This command is used to clear all the values on the panel so you can start fresh.
RESEt	Clears the “swap generated” message in the status field. This command is used to reset the status field after generating a Swap job. This status field is set on the panel to prevent a duplicate MOUNT statement for a previously generated volume being re-generated. If the previously generated Swap JCL is discarded and not used, the RESET command can be used to reset the status field allowing the MOUNT statement for the volume(s) to be re-generated.

To execute a command from the Menu “pull-down” list, place the cursor next to the desired command and press ENTER.

NOTE: The short forms of the these commands are shown by the underlined letters.

If you simply press ENTER, it checks to see if there are any swaps in progress on this system. If active swaps were found, they are automatically displayed, such as:

FDRPAS PLUG & SWAP – PANEL A.P – FORMAT 1

```
--------------------------- FDRPAS Plug & Swap  ------------- Row 1 to 2 of 2
COMMAND ===>                                                 SCROLL ===> PAGE
Menu                                                            Panel: 1 of 3
Command  Volume Unit              SMS    Swap to                Refresh 0
         Serial Addr        CU  Storage Offline
          Mask  Mask SSID Serial Group   Unit   Status                 09:39:27
-------  ------ ---- ---- ------ -------- ---- ---------------------------------
         HI17CF 17DF 9970 22398               SWAPPED
         HI17CE 17CE 9970 22398          17DE ACTIVE SWAP
         HI17CD 17CD 9970 22398          17DD ACTIVE SWAP
         HI17CC 17CC 9970 22398          17DC SYNCHRONIZING
```

You have several options:

- As long as there are no volumes displayed on the screen, you can simply press ENTER to display any FDRPAS swaps that are active on this system (either SWAP tasks or MONITOR tasks).
- You can also request that volumes that are not currently involved in a swap be added to the display; this can be useful to initiate swaps or to verify the current location of selected volumes. Details are later in this section.
- If there are already volumes displayed, type the “Active” command (or just “A”) on the Command line to add any additional active swaps to the display.
- If the display currently contains at least one volume that was added to the display because it was active (by pressing ENTER on a empty screen or by typing the “Active” command), then pressing ENTER again scans for active volumes again and adds any newly active swaps to the display.
- If the display currently contains only volumes that were added by specific request, then pressing ENTER simply updates the status of the displayed volumes, and does not check for active swaps.

RECOMMENDATION: If you just keep pressing ENTER, the panel displays all active swaps, and adds swaps that later became active. Completed swaps remain on the display. However, note that if a swap begins and ends between two presses of the ENTER key; then the panel never sees the active swap and does not add it to the display.

RECOMMENDATION: Alternately, you can type a volume serial prefix, a unit address prefix, an SSID, or an SMS storage group to display all of the volumes/devices selected, whether active or not. Details are later in this section. Pressing ENTER updates the status of the displayed volumes but does not automatically add any active swaps of other devices (unless you type the “ACTIVE” command). This is preferable when you are swapping all DASD volumes on a control unit (SSID) or in a range of addresses, since all those DASD volumes are constantly displayed. You can easily see which ones have active swaps, completed swaps, or have not been swapped.

THE THREE DISPLAY FORMATS

The ISPF dialog supports three display formats.

Display format 1, the default format, shown on the previous page, has one line of information about each volume. You can, of course, scroll up and down to view the complete list if it extends beyond one page.

Notice that in the upper right of the default format it says “Panel: 1 of 3”. This indicates that the default format (panel format 1) is in use.

You can switch display formats by using the PF11 and PF10 keys, or the “RIGHT” and “LEFT” commands. The FDRPAS dialogs remember what format you were using when you last exited from the dialog and use that format again when you enter the FDRPAS dialog again.

Pressing PF11 once selects display format 2, which uses two lines per volume, for example,

FDRPAS PLUG & SWAP – PANEL A.P – FORMAT 2

```
--------------------------- FDRPAS Plug & Swap  ------------- Row 1 to 2 of 2
COMMAND ===>                                                 SCROLL ===> PAGE
Menu                                                            Panel: 2 of 3
Command  Volume  Unit  Swap to                                  Refresh 0
         Serial  Addr  Offline
          Mask   Mask   Unit     Status                                09:49:53
-------  ------  ----  -------  ------------------------------------------------
         HI17CC  17CC   17CA     SYNCHRONIZING
          Pass:         % Trks to copy:         Copied:         Updated:
         HI17C2  17C2   17C1     WAIT FOR CONFIRMSWAP
          Pass: 8       % Trks to copy: 0       Copied: 0       Updated: 0
```

Pressing PF11 again selects display format 3, which uses five lines per volume, for example,

FDRPAS PLUG & SWAP – PANEL A.P – FORMAT 3

```
--------------------------- FDRPAS Plug & Swap  ------------- Row 1 to 2 of 2
COMMAND ===>                                                 SCROLL ===> PAGE
Menu                                                            Panel: 3 of 3
Command  Volume  Unit  Swap to                                  Refresh 0
         Serial  Addr  Offline
          Mask   Mask   Unit     Status                                09:44:59
-------  ------  ----  -------  ------------------------------------------------
         HI17CC  17CC   17CA     SYNCHRONIZING
         Pass:         % Trks to copy:         Copied:         Updated:
         Source - Reserve: 0  Level: 1  Pace:       Type: 3390    Cyls: 1113
         Target - Reserve: 1  Level: 1  Pace:       Type: 3390    Cyls: 1113
         Storgrp:          SSID: 9970  CU Serial#: 22398

         HI17C2  17C2   17C1     WAIT FOR CONFIRMSWAP
         Pass: 8       % Trks to copy: 0       Copied: 0       Updated: 0
         Source - Reserve: 0  Level: 1  Pace: 0     Type: 3390-3  Cyls: 3339
         Target - Reserve: 1  Level: 1  Pace: 0     Type: 3390-3  Cyls: 3339
         Storgrp:          SSID: 9970  CU Serial#: 22398
```

Pressing PF10 returns to formats 1 and 2.

Format 1 is shown in the examples in the rest of this section.

ADDING VOLUMES TO THE DISPLAY

You can add additional volumes to the display, selecting volumes by the volume serial, unit address, subsystem ID (SSID), control unit serial number, or SMS storage group. The last three can be used only if you are using a panel format that includes those fields.

On a blank line (or even a line that currently displays a volume if you no longer want to see it), tab to the field that you wish to use for selection and type the appropriate value. If a complete volume serial or unit address is entered, only that one volume is displayed. If you type an SMS storage group, SSID, or serial number, all volumes matching that value are displayed. The SSID or serial number is a convenient way to display all volumes in an existing DASD subsystem so that you can swap all or some of them to new hardware; it can also be used to verify when all volumes have been swapped off that subsystem.

For example, if you type a controller serial number (CU Serial):

FDRPAS PLUG & SWAP – PANEL A.P – ADDING VOLUMES TO THE DISPLAY

```
---------------------------- FDRPAS Plug & Swap  ------------- Row 1 to 1 of 1
COMMAND ===>                                                  SCROLL ===> PAGE
Menu                                                             Panel: 1 of 3
Command  Volume Unit              SMS    Swap to                  Refresh 0
         Serial Addr        CU  Storage Offline
          Mask  Mask SSID Serial Group   Unit   Status                  11:28:19
-------  ------ ---- ---- ------ -------- ---- ----------------------------------
'''                     LR221
```

and press ENTER.

FDRPAS PLUG & SWAP – PANEL A.P – VOLUMES ADDED TO THE DISPLAY

```
---------------------------- FDRPAS Plug & Swap  ----------- Row 1 to 34 of 59
COMMAND ===>                                                  SCROLL ===> PAGE
Menu                                                             Panel: 1 of 3
Command  Volume Unit              SMS    Swap to                  Refresh 0
         Serial Addr        CU  Storage Offline
          Mask  Mask SSID Serial Group   Unit   Status                  10:18:01
-------  ------ ---- ---- ------ -------- ---- ----------------------------------
         SCR081 3390 8102 LR221               INACTIVE
'''
         SCR089 22ED 8102 LR221               INACTIVE
…
```

It displays all volumes in the DASD subsystem with that SSID. As many as fit on the screen are displayed. You may need to scroll up and down (PF7 and PF8 or the UP and DOWN commands) to see the entire list.

Wild card characters can be used to select multiple volumes, units, storage groups, or control units. An asterisk (*) is the only supported wild card character. If it appears in the middle of a selection string, it represents exactly one character, while if it appears at the end it represents one or more characters.

For example,

- A volume serial number mask of **DB* selects all online volumes with serials of *xx*DB*xx*.
- A unit address mask of 12* selects all online volumes with addresses of 12*xx*.
- A storage group name of DB**3 selects all online volumes in SMS storage groups with names of DB*xx*3.
- An SSID of *3* selects all online volumes in subsystems with an SSID of *x*3*xx*.

ISPF line commands are supported for inserting and deleting entries in the display. In the "Command" column, next to any entry, type:

- **"D"** to delete a volume from the display.
- **"I"** to insert a blank entry in the display; it can be modified to add more volumes to the displayed volume list.

SORTING AND FILTERING THE DISPLAY

Type the SORT command on the COMMAND line to sort the displayed volumes and/or filter the display based on STATUS values. The SORT command displays this selection screen:

FDRPAS SORT – PANEL A.P – SORT COMMAND

```
---------------------------- FDRPAS Plug & Swap - Sort  --------------------
 COMMAND ===>                                                SCROLL ===> PAGE


 1) Specify the numeric sequence and order (A/D) of field(s) to sort:

    Field    Seq Order  Description
    -------- --- -----  ------------------
    VOL       _   _     Volume  Serial
    UNIT      _   _     Unit Address
    SSID      _   _     SSID
    STORGRP   _   _     Storage Group
    CU        _   _     Controller Serial
    DEVTYPE   _   _     Device Type
    OFFLINE   _   _     Offline Unit
    STATUS    _   _     Status

 2) Select the STATUS values of rows to be included in the display:

    Select   Status Value
    -------- ----------------------------
       S     ACTIVE
       S     BYPASS
       S     CANCEL
       S     FAIL
       S     INACTIVE
       S     SWAPPED
       S     SYNCHRONIZING
       S     WAIT FOR CONFIRM
       S     WAIT FOR CONSOLE REPLY
       S     WAIT FOR HYPERSWAP TO BE DISABLED
       S     WAIT FOR AUTOSWAP TO BE DISABLED
```

In the first section, place a digit in the SEQ field for each field on which the display is to be sorted; the field with SEQ of 1 is sorted first, then the field with SEQ of 2, and so on. Place an “A” (ascending) or “D” (descending) in the ORDER field to specify the sort order (“A” is assumed if not specified). Note that sorting on the STATUS field, ascending, places ACTIVE swaps first.

INNOVATION DATA PROCESSING suggests that sorting on STATUS (1) and VOL (2) provide a useful view of your volumes, especially if you have selected a range of volsers, device addresses, or an SSID to display.

In the second section, blank out the “S” in front of any values for STATUS that you do not want to see. Only those volumes with a status for which “S” is still present are displayed when you press END (PF3) to return to the status screen. If one or more values have the “S” blanked out, the status screen shows the text “*ROWS EXCLUDED*” to remind you that you have filtered the display.

The values you fill in on this screen are remembered in your ISPF profile and used the next time you re-enter the FDRPAS dialogs. To change them, you must use the SORT command again and modify the values.

MONITORING FDRPAS

When active swaps are displayed, you can monitor their progress. The display looks like:

FDRPAS PLUG & SWAP – PANEL A.P – MONITORING PROGRESS

```
--------------------------- FDRPAS Plug & Swap  ------------- Row 1 to 2 of 2
COMMAND ===>                                                 SCROLL ===> PAGE
Menu                                                            Panel: 1 of 3
Command  Volume Unit              SMS    Swap to                 Refresh 0
         Serial Addr         CU  Storage Offline
          Mask  Mask SSID Serial Group    Unit   Status                12:02:04
-------  ------ ---- ---- ------ -------- ---- ---------------------------------
         HI17CC 17CC 9970 22398           17CA ACTIVE SWAP   CONFIRMSWAP
         HI17C2 17C2 9970 22398           17C1 ACTIVE SWAP   CONFIRMSWAP   43%
```

The status of **ACTIVE** indicates that the swap copy is in progress. In this example, **CONFIRMSWAP** indicates that `CONFIRMSWAP=YES` was specified on the SWAP request; when the volumes are synchronized, the swap waits, copying newly updated tracks as necessary, until this panel is used to confirm that the swap is complete. If CONFIRMSWAP is not present, the swap is automatically completed when the volumes are synchronized.

The status area may contain other messages. For example, if the SWAP is waiting for a reply (message FDRW01, or FDRW68), it indicates so (for example, WAIT FOR CONSOLE REPLY FDRW68 CAUTION REQUEST). I/O ERROR ON TARGET indicates that `SWAPIOERR=RETRY` and an I/O error while copying tracks has caused the swap to retry the error periodically until resolved or aborted.

If the swap is active, the end of the status area shows the percent(%) of the tracks that need to be copied during this pass. If there were updates to the volume during the copy phase, the value goes to 100% and then resets to the percent(%) of the tracks currently being recopied.

If using panel 2 of 3, the second line shows the number of the copy pass that is currently executing (see "Phase 3: Copy" in Section 300.3), the percentage complete of the pass, the total number of tracks to copy in the pass, and the number of tracks already copied in the pass. "Updated" indicates the number of tracks that have been updated on the source volume during the current pass, on this system; these tracks (plus those found updated by other systems) are copied in the next pass.

If using panel 3 of 3, the next two lines show, for the source volume and the target device, the number of RESERVE macros that have been issued against them (the target always shows at least 1, issued by FDRPAS), the IOSLEVEL, and the current I/O pacing value. You can over type the target pacing value for any active volume to dynamically change the I/O pacing in use (see "PACEDELAY=" in Section 310.2 for details).

Press ENTER to update the information displayed. Alternately, over type the number after **REFRESH** with a value. The screen is updated automatically, every two seconds, until this value counts down to zero; the two second refresh interval can be changed on the FDRPAS options panel shown later in this section.

Once the volumes in a swap with `CONFIRMSWAP=YES` have been synchronized, the display looks like:

FDRPAS PLUG & SWAP – PANEL A.P – CONFIRMSWAP

```
--------------------------- FDRPAS Plug & Swap  ------------- Row 1 to 2 of 2
COMMAND ===> confirmswap                                     SCROLL ===> PAGE
Menu                                                            Panel: 1 of 3
Command  Volume Unit              SMS    Swap to                 Refresh 0
         Serial Addr         CU  Storage Offline
          Mask  Mask SSID Serial Group    Unit   Status                12:11:36
-------  ------ ---- ---- ------ -------- ---- ---------------------------------
         HI17CC 17CC 9970 22398           17CA WAIT FOR CONFIRMSWAP
         HI17C2 17C2 9970 22398           17C1 WAIT FOR CONFIRMSWAP
```

To confirm the swaps and cause FDRPAS to complete them, use the CONFIRMSWAP command (or just CO). You can:

- Type it in the "Command" column next to each volume to be confirmed.
- Type it on the "COMMAND" line at the top of the screen. This applies to every volume on the display (even those that may not be currently visible) that is in WAIT FOR CONFIRMSWAP status.

Once you enter the command, the display looks like:

FDRPAS – PANEL A.P – CONFIRMSWAP

```
---------------------------- FDRPAS Plug & Swap  ------------- Row 1 to 2 of 2
COMMAND ===>                                                  SCROLL ===> PAGE

Press enter to COnfirm the following selected volumes, or PF3 to return to the
prior panel.

Command  Volume Unit              SMS   Swap to
         Serial Addr        CU  Storage Offline
          Mask  Mask SSID Serial Group   Unit   Status
-------  ------ ---- ---- ------ -------- ---- --------------------------------
CONFIRM  HI17CC 17CC 9970 22398           17CA WAIT FOR CONFIRMSWAP
CONFIRM  HI17C2 17C2 9970 22398           17C1 WAIT FOR CONFIRMSWAP
```

Only the volumes to be confirmed are displayed. Press ENTER to complete the swap, or press PF3 (END) to ignore the confirmation. Note that the word "CONFIRM" is filled in on the command field for each volume; volumes can be blanked out before pressing ENTER to exclude them.

Once you confirm the swap, FDRPAS completes the swap and the display looks like:

FDRPAS PLUG & SWAP – PANEL A.P – COMPLETING THE SWAP

```
---------------------------- FDRPAS Plug & Swap  ------------- Row 1 to 2 of 2
COMMAND ===>                                                  SCROLL ===> PAGE
Menu                                                             Panel: 1 of 3
Command  Volume Unit              SMS   Swap to                   Refresh 0
         Serial Addr        CU  Storage Offline
          Mask  Mask SSID Serial Group   Unit   Status                12:19:01
-------  ------ ---- ---- ------ -------- ---- --------------------------------
         HI17CC 17CA 9970 22398              SWAPPED
         HI17C2 17C1 9970 22398              SWAPPED
```

The unit address, SSID, and serial number of the target device are now displayed.

This is also the format of the display for completed swaps when `CONFIRMSWAP=YES` is not used.

INITIATING SWAP TASKS

You can also initiate SWAP and MONITOR tasks from the FDRPAS ISPF panels. If multiple systems are involved, FDRPAS MONITOR tasks for the offline target volumes must be started on every system, before the SWAP task starts.

FDRPAS – PANEL A.P – INITIATING SWAP TASKS

```
---------------------------- FDRPAS Plug & Swap  ------------- Row 1 to 2 of 2
COMMAND ===> swap                                             SCROLL ===> PAGE
Menu                                                             Panel: 1 of 3
Command  Volume Unit              SMS    Swap to                  Refresh 0
         Serial Addr         CU  Storage Offline
          Mask  Mask SSID Serial Group   Unit   Status                  11:45:38
-------  ------ ---- ---- ------ -------- ---- --------------------------------
         HI17CC 17CC 9970 22398           17CA INACTIVE
         HI17C2 17C2 9970 22398           17C1 INACTIVE
```

To initiate a swap, display one or more inactive online volumes (not currently involved in a swap) as described earlier. In the column “Swap to Offline Unit”, fill in a 4-digit device address for the offline target volume that the online volume is to be swapped. When complete, type the SWAP command (or just SW) for a SWAP operation or DUMP (or just DU) for a SWAPDUMP operation. You can:

- Type it in the “Command” column next to each volume to be swapped as shown above.
- Type it on the “COMMAND” line at the top of the screen. This applies to every volume on the display (even those that may not be currently visible) that is in INACTIVE status with a target device filled in.
- Place the cursor on the SWAP command in the “pull-down” menu.

This creates one or more FDRPAS batch jobs for the requested swaps. See the description of the options on the next page to understand how the FDRPAS batch jobs are managed.

NOTE: If you are swapping many volumes, for best performance you should spread the SWAP tasks across several systems, so you may need to add JCL or control statements to route the jobs to the systems to do so. If multiple systems are involved, you must start MONITOR tasks on every system before initiating the SWAP task (you can start MONITOR tasks from the ISPF dialog or have the FDRPAS job submit the MONITOR tasks by using the //PASJOB DD statements).

HINT: If you are swapping volumes from one control unit to another, here is an easy way to initiate the swaps with a minimum of typing. On an empty line, type the address mask for the old control unit, for example, 17. On the same line, type the address mask of the new control unit in the “swap to” column, for example, 21**. It displays all the volumes that are still online in the old address range (1700-17CF), and propagates the target address of 21** to each entry. As explained under “SWAPUNIT=” in Section 310.3, FDRPAS swaps each 17xx DASD volume to the matching 21xx target volume.*

SETTING FDRPAS OPTIONS

You can set the options to be used during the SWAP or SWAPDUMP operation by typing the OPTIONS command beforehand to get this panel:

FDRPAS OPTIONS – PANEL A.P – OPTIONS COMMAND

```
--------------------------- FDRPAS Plug & Swap - Options  --------------------
COMMAND ===>                                                SCROLL ===> PAGE
                                                                 More:     +
Options for SWAP command:                Options for SWAPDUMP command:
CONFIRMSWAP  ===> NO     (yes no)        CONFIRMSPLIT ===> NO     (yes no)
CONFMESS     ===> NO     (yes no)        CONFMESS     ===> NO     (yes no)
LOGMESS      ===> YES    (yes no)        LOGMESS      ===> NO     (yes no)
PACEDELAY    ===> 0      1/100 seconds   PACEDELAY    ===> 0      1/100 seconds
SWAPDELAY    ===> 15     seconds         SWAPDELAY    ===> 15     seconds
CHECKTARGET  ===> NO                     CHECKTARGET  ===> NO
LARGERSIZE   ===> NO     (ok no)         LARGERSIZE   ===> NO     (ok no)
PACING       ===> STATIC (dynam static)  PACING       ===> STATIC (dynam static)
ALLOWPAV     ===> NO     (yes no)        ALLOWPAV     ===> NO     (yes no)

Other options for SWAP and SWAPDUMP commands:
Maxvols      ===> 1      maximum number of volumes to generate per process
Maxtasks     ===> 1      maximum number of concurrent volumes to process
Maxactive    ===> NO     limit the number of concurrent tasks in copy phase
Interval     ===> 2      refresh interval in seconds
WTOR         ===> YES    prompt console operator for message reply
PRINT=ALL    ===> NO     print all data set names and tracks copied

Options for MONITOR command:
MAXTASKS     ===> 64     maximum number of concurrent volumes to process
DYNMON       ===> NO     (yes no)
DURATION     ===>        minutes, blanks for no duration
LOGMESS      ===> NO     (yes no)
ALLOWPAV     ===> NO     (yes no)

Options for SIMSWAP and SIMSWAPMON commands:
CHECKSOURCE  ===> YES    check integrity of the source volume VTOC and VVDS

Options for CONFIRM command:
CONFMESS     ===> YES    (yes no)

Options for AUTOSWAP command:
CONGROUP   ===>          groupname

Job Statement Information:
  ===> //pfxA  JOB (acct),'name',NOTIFY=pfx
  ===> //*
  ===> //*
  ===> //*

  STEPLIB    ===> 'fdrpas.loadlib'

Dataset name containing Swap Exclude statements  (optional)
  Dsname     ===>
  Member     ===>                        (required for partitioned data set)
Job Statement Information for Monitor Jobs:

Monitor #1
       CPUID ===>                       (specify to include MONITOR in PASJOB)
  ===>
  ===>
…
```

You can over type any of the options shown on the previous panel. The values are saved

in your ISPF profile for use with all subsequent SWAP, SWAPDUMP, SIMSWAPMON, and SIMSWAP operations started by this ISPF user id.

The options on the first part of the panel correspond to options that can be specified on a SWAP,SWAPDUMP, SIMSWAPMON, or SIMSWAP statement. See Section 310.2 "SWAP, SWAPDUMP, SIMSWAP, SIMSWAPMON Statements" for a description of these options.

The options on the bottom ("Other Options"), except for the option "Interval", control the submission of swaps from the ISPF panel:

Maxvols=

***nnnn* –** Specifies the maximum number of volumes that the panel consolidates into a single FDRPAS SWAP task (address space) when you enter the SWAP or SWAPDUMP command and request that multiple volumes be swapped. It creates an FDRPAS batch job with one SWAP/SWAPDUMP statement and multiple MOUNT statements. If the number of volumes requested at one time is larger than **Maxvols**, then multiple FDRPAS batch jobs are created, with up to **Maxvols** volumes in each one. Each FDRPAS batch job processes its assigned volumes serially, one at a time, unless **Maxtasks** is greater than 1.

Default: 1 (one FDRPAS batch job per volume) and the maximum is 250.

Maxtasks=

***nn* –** Equivalent to the MAXTASKS=, operand on the SWAP and SWAPDUMP statement (see "MAXTASKS=" in Section 310.2). **Maxtasks** has no meaning unless **Maxvols** is set to a value larger than 1.

Default: 1 (process one volume at a time, serially) and the maximum is 64.

Maxactive=

***nnn* –** Equivalent to the MAXACTIVESWAPS= operand on the SWAP and SWAPDUMP statement (see "MAXACTIVESWAPS=" in Section 310.2).

WTOR=

Equivalent to the WTOR= operand on the SWAP and SWAPDUMP statement (see "WTOR=" in Section 310.2).

Interval=

***nn* –** The refresh interval, in seconds, used when you enter a value for **Refresh** on the main FDRPAS panel. Refresh counts down to zero, every **Interval** seconds and automatically refreshes the display. This allows you to monitor FDRPAS operations without constantly pressing ENTER.

Default: 2.

SUSPENDING AND RESUMING ACTIVE SWAPS

You can use the FDRPAS ISPF panel to temporarily suspend an active swap without terminating it and resume it later, if needed. While suspended, FDRPAS does not copy any tracks from the source volume to the target device, but it still monitors the source volume for updates. When resumed, FDRPAS again copies tracks.

Type the SUSPEND command (or just SU) in the “Command” column next to the active swap to be suspended. This can be done only on the system running the SWAP task (indicated by MAIN on the right hand). The status changes to SUSPEND. To resume, type the RESUME command (or just RE) next to any suspended swap.

FDRPAS – PANEL A.P – SUSPEND SWAP TASKS (PANEL 3 OF 3)

```
COMMAND ===>                                                  SCROLL ===> PAGE
Menu                                                             Panel: 3 of 3
Command  Volume  Unit  Swap to                                    REfresh 0
         Serial  Addr  Offline
          Mask   Mask   Unit     Status
-------  ------  ----  -------   ---------------------------------------------
suspend  SH20CC  20CC   21CC     ACTIVE                                 (MAIN)
         Pass: 1    89 % Tracks to copy: 1246   Copied: 1110   Updated: 2504
         Source - Reserve: 0   Level: 1   Pace: 0   Type: 3390-9  Cyls: 10017
         Target - Reserve: 1   Level: 1   Pace: 0   Type: 3390-9  Cyls: 10017
         Storgrp:          SSID: 0310  CU Serial#: 12345
```

You can also reduce the overhead of FDRPAS without totally suspending copy I/O by over typing the target device pacing value. This value is the number of 1/100 seconds to delay between copy I/O's (each copy I/O typically copies 15 tracks). A pacing value of 5 or 10 significantly reduces FDRPAS overhead while allowing the swap to continue; however, it takes longer.

TERMINATING ACTIVE SWAP TASKS

You can use the FDRPAS ISPF panel to terminate an active swap, if needed.

Type the ABORT command (or just AB) in the “Command” column next to the active swap to be aborted. This can be done on a system running the SWAP task or a system running a MONITOR task for the volume.

FDRPAS – PANEL A.P – TERMINATE ACTIVE SWAPS

```
COMMAND ===>                                                  SCROLL ===> PAGE
Menu                                                             Panel: 1 of 3
Command  Volume Unit              SMS   Swap to                  Refresh 0
         Serial Addr        CU  Storage Offline
          Mask  Mask SSID Serial Group   Unit   Status                 11:07:35
-------  ------ ---- ---- ------ -------- ---- ---------------------------------
         SH20CC 20CC 8100 76421              ACTIVE SWAP                 52%
abort    SH20CD 20CD 8100 76421              ACTIVE SWAP                 75%
         SH20CE 20CE 8100 76421              ACTIVE SWAP                 98%
```

REPLYING TO CONSOLE MESSAGES

The panel displays SWAP tasks that are waiting for FDRW01 or FDRW68 replies, and allows you to reply to the message from ISPF instead of the console.

This example shows one task waiting for FDRW01 (because CONFMESS=YES was specified) and another waiting for FDRW68 (because the proper number of MONITOR tasks has not responded).

FDRPAS – PANEL A.P – MESSAGE COMMAND

```
Command  Volume Unit              SMS   Swap to                   Refresh 0
         Serial Addr        CU  Storage Offline
          Mask  Mask SSID Serial Group   Unit   Status                 10:30:59
-------  ------ ---- ---- ------ -------- ---- -------------------------------
msg      SY4006 41C6 0300 ABXNH          41C4 WAIT FOR CONSOLE REPLY FDRW01
msg      SY4007 41C7 0300 ABXNH          41C5 WAIT FOR CONSOLE REPLY FDRW68
```

To reply, type the MSG command in the command area (or type MSG on the command line to invoke it for all swaps in the display that are waiting for message reply). You get this panel that shows the full text of each message. Type a reply in the appropriate area to reply to the desired messages.

FDRPAS – PANEL A.P – MESSAGE COMMAND RESPONSE

```
---------------------------- FDRPAS Plug & Swap  ------------- Row 1 to 2 of 2
COMMAND ===>                                                  SCROLL ===> PAGE

*** CAUTION *** The volumes in the FDRW68 message must either be offline in all
non-responding systems or an FDRPAS monitor must be started in those systems.
Do not reply "yes" without verification as this may result in severe problems.

The following operator console messages are waiting for reply, You may enter a
reply in this panel, or PF3 to return to the prior panel.
--------------------------------------------------------------------------------
Message:  FDRW01 CONFIRM REQUEST TO SWAP VOL=SY4006 TO UNIT=41C4 ON   3 SYSTEMS
REPLY YES OR NO
Reply:
--------------------------------------------------------------------------------
Message:  FDRW68 CAUTION REQUEST TO SWAP VOL=SY4007 TO UNIT=41C5 ON   1 SYSTEMS
IGNORING NON-RESPONDING CPUS REPLY YES,NO,RETRY
Reply:
--------------------------------------------------------------------------------
```

DISPLAYING SWAP HISTORY

If you type the HISTORY command (or just HI) on the command line of the FDRPAS panel, you can display FDRPAS history records, as shown above, for all volumes that have been successfully swapped by FDRPAS. See Section 300.6 "FDRPAS History Records" for a description of History records.

FDRPAS HISTORY – PANEL A.P – HISTORY COMMAND

```
---------------------------- FDRPAS Plug & Swap History  ----- Row 1 to 7 of 7
COMMAND ===>                                                 SCROLL ===> PAGE

Command  Volume  Unit  Swapped    System     Date       Time
         Serial  Addr  to Unit
-------  ------  ----  -------   --------  ----------  --------
         HI17CD  17CD  17CB      CPUB      07/26/2010  10:09:15
         HI17CD  17CD  17CB      CPUA      07/26/2010  10:09:16
         HI17CD  17CD  17CB      CPUC      07/26/2010  10:09:16
         PROD18  01C6  0200      CPUC      09/10/2010  09:22:05
         SMS802  22E9  22D9      CPUC      09/14/2010  18:14:08
         SMS802  22E9  22D9      CPUA      09/14/2010  18:14:08
         SMS802  22E9  22D9      CPUB      09/14/2010  18:14:08
```

The sample display above shows that volume HI17CD was swapped from address 17CD to 17CB on three systems, on the date and time displayed.

On the FDRPAS panel, you have several options for selecting the history records to be displayed:

- If no volumes are displayed on the panel, you can type `HISTORY` on the command line to display the most recent swap activity for every volume recorded in the history records. Alternately, you can type `HISTORY ALL` to display all swap activity for every volume recorded (for volumes that have been swapped more than once).
- If volumes are displayed on the panel, typing `HISTORY` or `HISTORY ALL` displays history only for the displayed volumes.
- If you type `HISTORY` followed by a volume serial number prefix, such as `HISTORY TSO`, on the command line, it displays history only for the volumes specified, regardless of whether volumes are displayed or not. You can also follow the prefix with the `ALL` option, such as `HISTORY TSO ALL`.

310.32 SAMPLE FDRPAS ISPF SESSION

This step-by-step procedure shows the use of the ISPF panels to obtain and create the job necessary to run a FDRPAS swap process.

In this example, we want to swap all the devices on the Control Unit (CU) Serial number that begins with “LR22”, so we type “LR22*” in the “CU Serial” field.

FDRPAS SAMPLE SESSION – ENTERING VOLUME SERIAL MASK

```
--------------------------- FDRPAS Plug & Swap  ------------- Row 1 to 1 of 1
COMMAND ===>                                                  SCROLL ===> PAGE
Menu                                                             Panel: 1 of 3
Command  Volume Unit              SMS   Swap to                  Refresh 0
         Serial Addr        CU  Storage Offline
          Mask  Mask SSID Serial Group   Unit   Status                12:39:54
-------  ------ ---- ---- ----- -------- ---- --------------------------------
'''                     LR22*
```

When we press ENTER, we see all the selected devices and the information about these devices.

FDRPAS SAMPLE SESSION – DISPLAYING ALL MATCHING DEVICES

```
--------------------------- FDRPAS Plug & Swap  ---------- Row 1 to 36 of 163
OMMAND ===>                                                  SCROLL ===> PAGE
Menu                                                             Panel: 1 of 3
Command  Volume Unit              SMS   Swap to                  Refresh 0
         Serial Addr        CU  Storage Offline
          Mask  Mask SSID Serial Group   Unit   Status                12:40:01
-------  ------ ---- ---- ----- -------- ---- --------------------------------
         SCR081 3390 8102 LR221               INACTIVE
'''
         SCR089 22ED 8102 LR221               INACTIVE
         SAFEC8 22EC 8102 LR221               INACTIVE
         SMS812 22EB 8102 LR221 SGMV80        INACTIVE
         SMS811 22EA 8102 LR221 SGMV80        INACTIVE
         SMS802 22E9 8102 LR221 SG802         INACTIVE
         SMS803 22E8 8102 LR221 SG803         INACTIVE
         XX22E7 22E7 8102 LR221               INACTIVE
         SCR082 22E6 8102 LR221               INACTIVE
         SCR28B 22E5 8102 LR221               INACTIVE
         SCR086 22E4 8102 LR221               INACTIVE
         SCR080 22E3 8102 LR221               INACTIVE
         SCR087 22E2 8102 LR221               INACTIVE
         SCR088 22E1 8102 LR221               INACTIVE
         SCR085 22E0 8102 LR221               INACTIVE
         SH22D8 22D8 8102 LR221               INACTIVE
         SH22D7 22D7 8102 LR221               INACTIVE
         SH22D6 22D6 8102 LR221               INACTIVE
         SH22D5 22D5 8102 LR221               INACTIVE
         SH22D4 22D4 8102 LR221               INACTIVE
         SH22D3 22D3 8102 LR221               INACTIVE
         SH22D2 22D2 8102 LR221               INACTIVE
         SH22D1 22D1 8102 LR221               INACTIVE
         SH22D0 22D0 8102 LR221               INACTIVE
         SH22C8 22C8 8102 LR221               INACTIVE
         SH22C7 22C7 8102 LR221               INACTIVE
         SH22C6 22C6 8102 LR221               INACTIVE
         SH22C5 22C5 8102 LR221               INACTIVE
…
```

NOTE: You can type values on any of the selection fields (Volume Serial Mask, Unit Addr Mask, SSID, CU Serial, or SMS Storage Group) and FDRPAS matches on all the selection field specified.

The devices we want to swap these volumes to have a UCB address of 7*xxx*. Thus, we type "7*" in the "Swap to Offline Unit" field to have the FDRPAS panels generate this for all the displayed values.

FDRPAS SAMPLE SESSION – FILLING IN THE TARGET VOLUMES

```
---------------------------- FDRPAS Plug & Swap  ----------- Row 1 to 36 of 163
COMMAND ===>                                                 SCROLL ===> PAGE
Menu                                                            Panel: 1 of 3
Command  Volume Unit              SMS   Swap to                  Refresh 0
         Serial Addr        CU  Storage Offline
          Mask  Mask SSID Serial Group   Unit   Status                12:40:08
-------  ------ ---- ---- ------ -------- ---- --------------------------------
         SCR081 3390 8102 LR221          7*   INACTIVE
'''
         SCR089 22ED 8102 LR221               INACTIVE
         SAFEC8 22EC 8102 LR221               INACTIVE
         SMS812 22EB 8102 LR221 SGMV80        INACTIVE
         SMS811 22EA 8102 LR221 SGMV80        INACTIVE
         SMS802 22E9 8102 LR221 SG802         INACTIVE
         SMS803 22E8 8102 LR221 SG803         INACTIVE
         XX22E7 22E7 8102 LR221               INACTIVE
         SCR082 22E6 8102 LR221               INACTIVE
         SCR28B 22E5 8102 LR221               INACTIVE
         SCR086 22E4 8102 LR221               INACTIVE
         SCR080 22E3 8102 LR221               INACTIVE
         SCR087 22E2 8102 LR221               INACTIVE
         SCR088 22E1 8102 LR221               INACTIVE
         SCR085 22E0 8102 LR221               INACTIVE
…
```

The target devices are generated by using the mask and the suffix of the online unit addresses. Once generated, comparisons of the source and target devices are performed as well as the status of the target volumes. In this sample, you can see that some target devices were not found and not offline. There were differences in the sizes of the source and target volumes.

FDRPAS SAMPLE SESSION – DISPLAYING THE TARGET VOLUME DEVICES

```
---------------------------- FDRPAS Plug & Swap  ----------- Row 1 to 36 of 162
COMMAND ===>                                                 SCROLL ===> PAGE
Menu                                                            Panel: 1 of 3
Command  Volume Unit              SMS   Swap to                  Refresh 0
         Serial Addr        CU  Storage Offline
          Mask  Mask SSID Serial Group   Unit   Status                12:40:15
-------  ------ ---- ---- ------ -------- ---- --------------------------------
         SCR081 3390 8102 LR221          7390 INACTIVE - TARGET NOT FOUND
         SCR089 22ED 8102 LR221          72ED INACTIVE
         SAFEC8 22EC 8102 LR221          72EC INACTIVE - 003339 TO 010017 CYLS
         SMS812 22EB 8102 LR221 SGMV80   72EB INACTIVE - 002226 TO 003339 CYLS
         SMS811 22EA 8102 LR221 SGMV80   72EA INACTIVE - 002226 TO 003339 CYLS
         SMS802 22E9 8102 LR221 SG802    72E9 INACTIVE
         SMS803 22E8 8102 LR221 SG803    72E8 INACTIVE
         XX22E7 22E7 8102 LR221          72E7 INACTIVE - TARGET NOT OFFLINE
         SCR082 22E6 8102 LR221          72E6 INACTIVE - TARGET NOT OFFLINE
         SCR28B 22E5 8102 LR221          72E5 INACTIVE
         SCR086 22E4 8102 LR221          72E4 INACTIVE
         SCR080 22E3 8102 LR221          72E3 INACTIVE
         SCR087 22E2 8102 LR221          72E2 INACTIVE
         SCR088 22E1 8102 LR221          72E1 INACTIVE
         SCR085 22E0 8102 LR221          72E0 INACTIVE
…
```

Some of the target devices may not exist, so we can manually change them to the desired devices and vary the volumes offline after ensuring that they are the desired target devices. Any other discrepancies can be fixed here before generating the JCL.

FDRPAS SAMPLE SESSION – CHANGING TARGET UNITS

```
---------------------------- FDRPAS Plug & Swap  ----------- Row 1 to 36 of 162
COMMAND ===>                                                 SCROLL ===> PAGE
Menu                                                            Panel: 1 of 3
Command  Volume Unit              SMS   Swap to                  Refresh 0
         Serial Addr        CU  Storage Offline
          Mask  Mask SSID Serial Group   Unit   Status                 12:40:56
-------  ------ ---- ---- ------ -------- ---- --------------------------------
         SCR081 3390 8102 LR221          72EE INACTIVE
         SCR089 22ED 8102 LR221          72ED INACTIVE
         SAFEC8 22EC 8102 LR221          72EC INACTIVE - 003339 TO 010017 CYLS
         SMS812 22EB 8102 LR221 SGMV80   72EB INACTIVE - 002226 TO 003339 CYLS
         SMS811 22EA 8102 LR221 SGMV80   72EA INACTIVE - 002226 TO 003339 CYLS
         SMS802 22E9 8102 LR221 SG802    72E9 INACTIVE
         SMS803 22E8 8102 LR221 SG803    72E8 INACTIVE
         XX22E7 22E7 8102 LR221          72E7 INACTIVE
         SCR082 22E6 8102 LR221          72E6 INACTIVE
         SCR28B 22E5 8102 LR221          72E5 INACTIVE
         SCR086 22E4 8102 LR221          72E4 INACTIVE
         SCR080 22E3 8102 LR221          72E3 INACTIVE
         SCR087 22E2 8102 LR221          72E2 INACTIVE
         SCR088 22E1 8102 LR221          72E1 INACTIVE
         SCR085 22E0 8102 LR221          72E0 INACTIVE
…
```

Since we are moving some smaller devices to larger devices, we want to set the `LARGERSIZE=` option to `OK`.

FDRPAS SAMPLE SESSION – SETTING THE SWAP OPTIONS

```
---------------------------- FDRPAS Plug & Swap - Options  --------------------
COMMAND ===>                                                 SCROLL ===> PAGE

                                                                    More:     +
Options for SWAP command:                 Options for SWAPDUMP command:
CONFIRMSWAP  ===> NO      (yes no)        CONFIRMSPLIT ===> NO      (yes no)
CONFMESS     ===> NO      (yes no)        CONFMESS     ===> NO      (yes no)
LOGMESS      ===> YES     (yes no)        LOGMESS      ===> NO      (yes no)
PACEDELAY    ===> 0       1/100 seconds   PACEDELAY    ===> 0       1/100 seconds
SWAPDELAY    ===> 15      seconds         SWAPDELAY    ===> 15      seconds
CHECKTARGET  ===> YES                     CHECKTARGET  ===> NO
LARGERSIZE   ===> OK      (ok no)         LARGERSIZE   ===> NO      (ok no)
PACING       ===> STATIC  (dynam static)  PACING       ===> STATIC  (dynam static)
```

The recommended setting for the `DYNMON` operand in the MONITOR tasks is `YES`.

FDRPAS SAMPLE SESSION – SETTING THE MONITOR OPTIONS

```
---------------------------- FDRPAS Plug & Swap - Options  --------------------
COMMAND ===>                                                 SCROLL ===> PAGE

                                                                    More:   - +
Options for MONITOR command:
MAXTASKS     ===> 64     maximum number of concurrent volumes to process
DYNMON       ===> YES    (yes no)
DURATION     ===> 100    minutes, blanks for no duration
LOGMESS      ===> NO     (yes no)
```

The FDRPAS options need to be changed to add a STEPLIB since FDRPAS should always run from a STEPLIB. Additionally, the MONITOR tasks can be generated by supplying the necessary information to route the MONITOR tasks to the LPAR. We are typing that information here.

FDRPAS SAMPLE SESSION – ENTERING STEPLIB AND JOB STATEMENTS FOR MONITOR TASKS

```
---------------------------- FDRPAS Plug & Swap - Options  --------------------
COMMAND ===>                                                  SCROLL ===> PAGE

                                                                       More:   - +
  STEPLIB    ===> 'fdrpas.loadlib'

Dataset name containing Swap Exclude statements  (optional)
  Dsname     ===>
  Member     ===>                        (required for partitioned data set)

Job Statement Information for Monitor Jobs:

Monitor #1
       CPUID ===> 0912342818           (specify to include MONITOR in PASJOB)
  ===> //PASMONK  JOB (acct),'name',MSGLEVEL=(1,1),
  ===> //             MSGCLASS=X,CLASS=M
  ===> /*ROUTE    XEQ JESCPUA
  ===>
Monitor #2
       CPUID ===> 0212342818           (specify to include MONITOR in PASJOB)
  ===> //PASMONL  JOB (acct),'name',MSGLEVEL=(1,1),
  ===> //             MSGCLASS=X,CLASS=M
  ===> /*ROUTE    XEQ JESCPUB
  ===>
Monitor #3
       CPUID ===> 0112342818           (specify to include MONITOR in PASJOB)
  ===> //PASMONM  JOB (acct),'name',MSGLEVEL=(1,1),
  ===> //             MSGCLASS=X,CLASS=M
  ===> /*ROUTE    XEQ JESCPUC
  ===>
…
```

If you are going to use the ISPF panels to generate the swap jobs and you want to create the `//PASJOB` control statements, specify the CPUID for as many systems that you need to monitor. Specify the appropriate JOB, CLASS, and ROUTE control statements to get the job to the correct CPU for each monitor.

Displaying the generated JCL for the SIMSWAPMON process. After verifying the JCL and control statements, the job can be submitted.

FDRPAS SAMPLE SESSION – DISPLAY THE GENERATED JCL

```
  File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
EDIT------ JSB.SPFTEMP1.CNTL---------------------------- Columns 00001 00072
Command ===> submit                                       Scroll ===> CSR
****** ***************************** Top of Data ******************************
000001 //pfxT  JOB (acct),'user',NOTIFY=pfx,CLASS=A
000002 //*
000003 //*
000004 //*
000005 //FDRPAS   EXEC PGM=FDRPAS,REGION=0M
000006 //STEPLIB  DD  DISP=SHR,DSN=fdrpas.loadlib
000007 //SYSPRINT DD  SYSOUT=*
000007 //SYSUDUMP DD  SYSOUT=*
000009 //SYSIN    DD  *
000010   SIMSWAPMON  TYPE=FULL,
000011         CONFIRMSWAP=NO,
000012         CONFMESS=NO,
000013         LOGMESS=YES,
000014         PACEDELAY=0,
000015         SWAPDELAY=15,
000016         CHECKSOURCE=YES,
000017         CHECKTARGET=YES,
000018         LARGERSIZE=OK,
000019         PACING=STATIC,
000020         MAXTASKS=1,
000021         MAXACTIVESWAPS=NO,
000022         WTOR=YES
000023   MOUNT VOL=SCR081,SWAPUNIT=72EE
000024   MOUNT VOL=SCR089,SWAPUNIT=72ED
000025   MOUNT VOL=SAFEC8,SWAPUNIT=72EC
000026   MOUNT VOL=SMS812,SWAPUNIT=72EB
000027   MOUNT VOL=SMS811,SWAPUNIT=72EA
000028   MOUNT VOL=SMS802,SWAPUNIT=72E9
000029   MOUNT VOL=SMS803,SWAPUNIT=72E8
000030   MOUNT VOL=XX22E7,SWAPUNIT=72E7
000031   MOUNT VOL=SCR082,SWAPUNIT=72E6
000032   MOUNT VOL=SCR28B,SWAPUNIT=72E5
000033   MOUNT VOL=SCR086,SWAPUNIT=72E4
000034   MOUNT VOL=SCR080,SWAPUNIT=72E3
000035   MOUNT VOL=SCR087,SWAPUNIT=72E2
…
000186 //PASJOB    DD  DATA,DLM=ZZ
000187 //*CPUID=0912342818
000188 //PASMONU  JOB (acct),'user',MSGLEVEL=(1,1),
000189 //            MSGCLASS=X,CLASS=M
000190 /*ROUTE    XEQ JESCPUA
000191 //FDRPAS   EXEC PGM=FDRPAS,REGION=0M
000192 //STEPLIB  DD  DISP=SHR,DSN=fdrpas.loadlib
000193 //SYSPRINT DD  SYSOUT=*
000194 //SYSUDUMP DD  SYSOUT=*
000195 //SYSIN    DD  *
000196   MONITOR TYPE=SWAP,
000197           MAXTASKS=64,
000198           DYNMON=YES,
…
000365 //*CPUID=0212342818
000366 //PASMONV  JOB (acct),'user',MSGLEVEL=(1,1),
…
```

This display shows the status of the devices after submitting the SIMSWAPMON job.

FDRPAS SAMPLE SESSION – STATUS AFTER SUBMITTING SIMSWAPMON JOB

```
---------------------------- FDRPAS Plug & Swap  ----------- Row 1 to 15 of 15
COMMAND ===>                                                 SCROLL ===> PAGE
Menu                                                            Panel: 1 of 3
Command  Volume Unit              SMS   Swap to                  Refresh 0
         Serial Addr        CU  Storage Offline
          Mask  Mask SSID Serial Group   Unit   Status                 12:41:25
-------  ------ ---- ---- ----- -------- ---- ----------------------------------
         SCR081 3390 8102 LR221          72EE SYNCHRONIZING
         SCR089 22ED 8102 LR221          72ED SYNCHRONIZING
         SAFEC8 22EC 8102 LR221          72EC SYNCHRONIZING
         SMS812 22EB 8102 LR221 SGMV80   72EB SYNCHRONIZING
         SMS811 22EA 8102 LR221 SGMV80   72EA SYNCHRONIZING
         SMS802 22E9 8102 LR221 SG802    72E9 SYNCHRONIZING
         SMS803 22E8 8102 LR221 SG803    72E8 SYNCHRONIZING
         XX22E7 22E7 8102 LR221          72E7 SYNCHRONIZING
         SCR082 22E6 8102 LR221          72E6 SYNCHRONIZING
         SCR28B 22E5 8102 LR221          72E5 SYNCHRONIZING
         SCR086 22E4 8102 LR221          72E4 SYNCHRONIZING
         SCR080 22E3 8102 LR221          72E3 SYNCHRONIZING
         SCR087 22E2 8102 LR221          72E2 SYNCHRONIZING
         SCR088 22E1 8102 LR221          72E1 SYNCHRONIZING
         SCR085 22E0 8102 LR221          72E0 SYNCHRONIZING
…
```

If you see "**SYNCHRONIZING**" in the Status field for a while, the MONITOR tasks may be dynamically adding the SWAP units or one or more MONITOR tasks may not be running.

320 FDRPAS SPECIAL CONSIDERATIONS

320.1 FDRPAS SPECIAL HARDWARE CONSIDERATIONS

This chapter documents special hardware considerations for the use of FDRPAS. It is as complete as possible and contains all considerations known to INNOVATION DATA PROCESSING at the time of publication. However, there may be other considerations that have not been identified or that were discovered after publication.

This section should be reviewed carefully before performing any FDRPAS operations.

For the latest updates to hardware considerations, go to the INNOVATION DATA PROCESSING web site at: http://fdr.com. Click the "FTP Login" link, then type your site access code to login to the registered FTP site.

RELOCATING A DATA CENTER

FDRPAS can be used to relocate a data center by duplicating all of the online volumes in new DASD subsystems at the new site. Contact INNOVATION DATA PROCESSING for additional documentation with details on the procedures and considerations.

PREPARING THE TARGET DEVICES

The target devices should be varied offline to all system images. If the target device is not offline on an LPAR that a MONITOR task is running, special checking is done by the MONITOR task to ensure that this device is the same target device as specified by the main FDRPAS process and that the device is inactive on the LPAR this MONITOR task is running on. If so, then this volume is varied offline by this MONITOR task.

However, **you must not** mark the target devices as offline in your I/O configuration. If this is done, the device would be offline at the next IPL and the operating system would not find the volume at its new location.

You do not need to initialize the target devices in any way. FDRPAS is not sensitive to the contents of the target devices. However, you can initialize them if you like; this prevents annoying messages at IPL time. Also, if you specify the `CHECKTARGET=YES` operand, the target must be initialized but empty or never used (no valid volume label).

QUERY HOST ACCESS SUPPORT

Query Host Access (QHA) is supported and utilized for the storage subsystems that support this function. This function gives FDRPAS a list of all the LPARs that have the volume online, allowing FDRPAS to ensure that monitor tasks are running on all these LPARs. With this support, any SWAP task `EXCLUDE CPUID=` statements (see Section 310.4 "SWAP Task EXCLUDE Statement") are ignored on storage subsystems that support QHA. If you feel that you have a need to use an `EXCLUDE CPUID=` statement on such a subsystem, please contact INNOVATION Technical Support.

Query Host Access is available for the DS8870 in code Release 7.1, GA 6/07/2013, and we expect that QHA will soon become available for the DS8700 and DS8800. We urge all FDRPAS customers to install these microcode levels when available, to support enhanced validation by FDRPAS.

NOTE: Most EMC storage subsystems already provide comparable support, as discussed in Multi-System Determination.

MULTI-SYSTEM DETERMINATION

When your source volumes are in an IBM RVA, an early Oracle StorageTek SVA or any others that emulate a 3990-3, the #SYSTEMS= operand must be specified on the SWAP statement, because those systems do not allow FDRPAS to determine the number of systems with access to DASD in them. **Contact INNOVATION DATA PROCESSING before using the #SYSTEMS operand.**

When your source volumes are in an EMC Symmetrix subsystem (except the 4*xxx* series), FDRPAS can determine which LPARs actually have a source volume online. On these subsystems, any SWAP task EXCLUDE CPUID= statements (see SWAP Task EXCLUDE Statement) are ignored. If you feel that you have a need to use an EXCLUDE CPUID= statement on such a subsystem, please contact INNOVATION Technical Support.

When your source volumes are in an IBM 3990-6, IBM 2105 / IBM 2107 / DS6000 / DS8000 and others that emulate these, except for IBM DS8*xxx* subsystems with the Query Host Access feature (see Query Host Access Support), FDRPAS can determine how many system images can access the subsystem, but cannot determine which volumes are online on each subsystem. If the FDRPAS source volume is online to some systems or LPARs but is offline or not in the I/O configuration to others with a running z/OS system, and those other systems have some devices in that subsystem defined, FDRPAS expects that system to participate in the swap. If the target device is accessible by that system, and an FDRPAS MONITOR task is monitoring the target on that system, FDRPAS automatically determines if it should participate or not. Even if the target device is not accessible on a given system but that system is connected to the swapping system with GRS or MIM, FDRPAS determines this as long as a MONITOR task is running on that system.

In some environments, FDRPAS may identify some systems that can access a given DASD volume but are not able to participate in an FDRPAS swap. Since FDRPAS does not know they are unable to participate, they can result in an FDR234 REASON=M message and an FDRW68 message indicating non-responding systems. Possible causes include:

- LPARs that are currently idle, not running an operating system. This may mean that the LPAR has never been IPL'd or the operating system has been shutdown. For LPARs, this means that the LPAR is idle but has not been deactivated on the Hardware Management Console (HMC). In some cases, even deactivating an LPAR from the Hardware Management Console (HMC) does not remove it from the systems reported by the hardware.
- LPARs that are running z/VM but that are not running a z/OS-type guest operating system under z/VM, and are not running an FDRPASVM monitor.
- LPARs that are running a non-z/OS-type operating system, such as Linux or VSE.
- LPARs where the FDRPAS MONITOR task has a low priority, or LPARs that have a low priority, may prevent the MONITOR task from responding in time. INNOVATION DATA PROCESSING recommends that you reply "RETRY" to the FDRW68 message at least once to allow such systems time to respond. **You should always contact INNOVATION DATA PROCESSING before responding "YES" to this message.**

These systems usually are not using the z/OS volumes that you are swapping, so this is not a problem (if they do use the volumes, be sure to place them offline to those systems before the swap). The LPARs that have access to a DASD volume can be determined by using the SIMSWAP or SIMSWAPMON command of FDRPAS.

If you have LPARs that do not participate, you can address them in several ways, in order of preference:

1. Add EXCLUDE CPUID= statements in the SWAP task input for each of the non-participating systems
2. Reply "**YES**" to the FDRW68 message after insuring that all the systems that participate have been correctly identified by FDR233 messages. **However, this is not recommended. We suggest that you reply "NO" to terminate the swap,**

investigate the cause, update the EXCLUDE statements and resubmit the swap.

NOTE: If you code `NONRESPONDING=FAIL` on the SWAP statement, FDRPAS automatically replies "NO" to the FDRW68 message and fails the swap.

On some DASD subsystems, it is possible to configure them to emulate either an IBM 3990-3 or a 3990-6 control unit. If they are in 3990-3 mode, they may not support the commands FDRPAS uses to determine the number of systems. The console command:

```
DS QD,devnum
```

displays the real or emulated control unit type. If in doubt, try executing FDRPAS SIMSWAP or SIMSWAPMON without `#SYSTEMS=`. If the DASD volume does not support the proper commands, SIMSWAP/SIMSWAPMON tells you.

STARTING AND STOPPING SYSTEMS DURING SWAPS

If possible, you should avoid shutting down or IPLing systems or LPARs while FDRPAS swaps are running. At the very least, you should make arrangements so that the person running FDRPAS is notified of any scheduled or unscheduled shutdowns or IPLs.

If a system has an unscheduled shutdown, such as a system crash or hardware failure, any FDRPAS swaps that were running at the time usually fail with no harm done, if the failed system was participating. When the SWAP task does not get the required responses from the MONITOR task on the failed system, it prints a diagnostic message and fails the swap. If the failure occurs just at the last step of a swap, the swap may be successful and the failed system sees the volume on the new device when it is re-IPL'd.

If a system has a scheduled shutdown, then they need to `CANCEL (C)` or `STOP (P)` the FDRPAS job on that system; this normally lets any active swaps complete before terminating. If a MONITOR task is forced to terminate, then active swaps fail cleanly as described above.

If a system was not active when a swap started, but is re-IPL'd during the swap and puts the volume involved online, there is an issue. There is no FDRPAS MONITOR task running on that system, so it does not participate in active swaps. If swaps complete before a MONITOR task can be started on the new system, the system does not know about the swap and still tries to use the volume on the old device; if this occurs, contact INNOVATION DATA PROCESSING for assistance. If a MONITOR task is started before the swap terminates, FDRPAS recognizes that a system came in late and terminates the SWAP cleanly.

If you have used EXCLUDE statements, you should probably terminate any active swaps, update the parameters to account for the stopped or started system, and resubmit the SWAP.

SWITCHING CABLES AND CONFIGURING CHPIDS DURING SWAPS

Switching, plugging and unplugging cables, and configuring CHPIDs (CONFIG CHP command), on devices that FDRPAS is currently swapping is not recommended because in extremely rare circumstances this may cause a volume to swap successfully in some systems, but to fail in other systems (the swap failure is accompanied by error messages FDR243 and FDR244). IBM issued APAR OA27065 for z/OS 1.10 to fix the swap failure caused by the CONFIG CHP command.

IBM 2105/2107 AND DS6000/DS8000 HARDWARE CONSIDERATIONS

- If a source volume is in an IBM 2105 ESS with FICON channels, you should be at microcode level 1.5.2.114 or above so that FDRPAS can properly identify the attached systems. This does not affect target volumes but this microcode level is recommended even for target systems.
- If you are swapping from a 2105, 2107, DS6000, or DS8000 DASD volume to a another DASD volume, FDRPAS turns off feature bits in the Device Characteristics Extension (DCE) of the UCB of the source volume for all features that are not supported by the target device. Any IBM software that was using any of these features should stop using them so that they do not cause errors when the swap to the new device is completed. These features currently include: FlashCopy, Prefix CCW, Read Track Data CCW, Write Full Track CCW, Write Track Data CCW, Locate Record Erase CCW, and Prestage Trackset CCW.
- **IBM FlashCopy**: During a swap, the source volume cannot be used as the target of a FlashCopy, since FDRPAS has no way of knowing that the source tracks are being updated. FDRPAS disables FlashCopy V2 (data set flash) in the hardware for the source volume during the swap, so that any attempt to use it fails. Most products that implement FlashCopy (including FDRCOPY) automatically use normal read/write I/O when FlashCopy is not available.
- **PAV**: Parallel Access Volume (PAV) is supported by FDRPAS. FDRPAS dynamically disables Parallel Access Volume (PAV) on the source and target devices during the swap. By default, FDRPAS performs the PAV disable at the beginning of the swap operation. (The default can be changed to by specifying the `ALLOWPAV=YES` operand on the MONITOR and SWAP/SWAPDUMP statements.) If you are swapping from one DASD device with PAV to another, PAV is re-enabled after the swap. However, if you are swapping from a DASD volume that does not have PAV to a DASD volume with PAV or vice-versa, PAV is disabled on the PAV device until the next time you IPL; this is an IBM limitation because of fields that exist only in the UCB of a PAV device. There is a circumvention: if you update your I/O configuration so that the non-PAV source volumes are defined as type 3390B (PAV base), then FDRPAS can enable PAV when you swap the volume to a PAV-capable device; if the target device has `WLMPAV=YES` (dynamic PAV) then the source volume should also be defined with `WLMPAV=YES`. IBM says that it is permissible to use device type 3390B for non-PAV DASD volumes, it causes no harm. However, it requires an IPL or dynamic ACTIVATE to activate the new configuration before you do any swaps.

EMC SYMMETRIX HARDWARE CONSIDERATIONS

- EMC Symmetrix TimeFinder commands and EMC SnapShot-compatible commands should not be issued to volumes involved in an FDRPAS swap. These commands may fail or they may update the source volume in a way that FDRPAS cannot detect.
- **IBM-compatible PAV**: If you are using IBM-compatible Parallel Access Volume (PAV)s in 2105-emulation mode, then see the notes on IBM PAV above.
- If you have job streams that execute EMC utilities or other software that depends on special functions of the EMC Symmetrix system (such as TimeFinder) against volumes in a Symmetrix, and you use FDRPAS to swap those volumes to other hardware that does not support those functions (such as a subsystem from another vendor), you need to update those job streams to eliminate or replace that software.
- See “Duplex Copy” in Section 320.1 for SRDF considerations.

EMC CONSISTENCY GROUPS

FDRPAS supports EMC Consistency Groups. When the source volume is an EMC DASD volume, FDRPAS issues a hardware query to see if it is part of a consistency group. If so, it issues the same query against the target device. Unless both devices are EMC DASD volumes in a consistency group, the SWAP fails with message FDR234 REASON=O.

Then FDRPAS invokes an EMC API to determine if both the source and target are in the SAME consistency group. If not, the swap fails with message FDR234 REASON=O.

Therefore, FDRPAS allows a volume in a consistency group to be swapped only to another volume in the same consistency group. This check is made by the FDRPAS SWAP task, so if you are not running the consistency group software on every system, you must run the SWAP task on a system where it is running.

Before the swap, you need to update the group definition to add the FDRPAS offline target device by device address and refresh the group to include it. After the swap, since DASD volumes are usually added to consistency groups by volume serial number or SMS storage group, you may be able to remove the device address since the volume is now on the target device. You should also set the Consistency Group option `AUTO_REFRESH=YES` so that the group is automatically refreshed after FDRPAS swaps a volume in the group.

If you are swapping an EMC volume to a DASD volume in a non-EMC subsystem or to an EMC subsystem that cannot participate in an appropriate SRDF session, you should disable the consistency group before doing the swap, since consistency is not maintained after the swap.

Because the EMC consistency group software and FDRPAS use some of the same interfaces for monitoring I/O, INNOVATION DATA PROCESSING does not recommend starting or stopping the EMC software, or disabling or enabling consistency groups, while FDRPAS swaps are running, unless you are certain they do not affect the same devices. Otherwise, FDRPAS swaps may fail and the EMC software may generate error messages; however, no harm is done to your system.

If the EMC consistency group software library is not in the system linklist, you may need to specify that library as a STEPLIB in the FDRPAS SWAP task so that FDRPAS can invoke the proper EMC API module.

ORACLE STORAGETEK SVA AND IBM RVA HARDWARE CONSIDERATIONS

- **SnapShot Copy**: If a SnapShot copy request copying data to the source volume is detected during an FDRPAS swap operation, the FDRPAS intercepts mark the snapped tracks as updated and re-copy them during the next Phase 3 pass.
- Since SnapShot copy requires multiple I/O operations to the input and output devices, a SnapShot issued just at the moment that FDRPAS is in the final swap phase on one of the devices the device may fail. Also, if a program has previously determined that two DASD volumes are capable of SnapShot, but the FDRPAS has since swapped the output device to an ineligible device, the program may issue a SnapShot request that fails.
- If you have jobs that execute IBM or Oracle StorageTek utilities or other software that depends on special functions of the RVA or SVA system (such as SnapShot), and you use FDRPAS to swap those volumes to other hardware that does not support those functions (such as a subsystem from another vendor), you need to update those jobs to eliminate or replace that software.
- On RVA and early SVA systems, FDRPAS is unable to determine the number of systems that have access to the source volume. You must specify the #SYSTEMS= operand on the SWAP statement for such volumes. **Contact INNOVATION DATA PROCESSING before using the #SYSTEMS= operand.**

HITACHI (HDS) HARDWARE CONSIDERATIONS

Customers swapping volumes to a Hitachi 9*xxx* DASD subsystem must ensure that it is running at microcode level 01-13-19/00 or higher. At lower microcode levels, FDRPAS MONITOR tasks are not able to recognize that a swap is starting.

Customers swapping from Hitachi subsystems that emulate IBM 3990-6 control units should note: FDRPAS may not be able to determine all of the systems with access to the source volume. Hitachi supports more connections than a 3990, so in 3990 emulation the subsystem may not be able to report to FDRPAS all of the logical paths to the source volume, and FDRPAS may be unaware of some attached systems. To check execute the FDRPAS SIMSWAP function and verify that all expected attached systems are reported. If not, contact INNOVATION DATA PROCESSING for a circumvention. This is not an issue if the Hitachi subsystem is in IBM 2105 emulation.

If FDRPAS source DASD volumes are in a Hitachi subsystem that emulates an IBM 3990-3 control unit, FDRPAS is unable to identify the attached systems, so you must use the `#SYSTEMS=` operand on the SWAP statement for such volumes. Note that this refers to the 3990-3 control unit, not the 3390-3 DASD volume model; FDRPAS is not sensitive to the model of DASD volume emulated.

Customers using Hitachi ShadowImage (that uses PPRC) should read the notes "Duplex Copy" in Section 320.1.

MIDAW SUPPORT

On an IBM System z processor (and any successor processors), the I/O subsystem supports a channel programming construct called a Modified Indirect Addressing Word (MIDAW). MIDAWs allow for more efficient data transfer in some circumstances. To support MIDAWs, the operating system must be z/OS 1.7 (or beyond) or z/OS 1.6 with enabling PTFs.

All IBM DASD subsystems support MIDAWs but some non-IBM subsystems do not. IBM does not allow a swap between a DASD volume that supports MIDAWs and one that does not, or vice versa, so FDRPAS checks that the MIDAW capabilities of the source and target DASD volumes in a swap match. Both must support MIDAWs, or both must not, or the swap fails with an FDR234 REASON=5 error message.

If you are running FDRPAS (SWAP or MONITOR task) on a processor and z/OS that supports MIDAWs, you can query the overall status of MIDAWs with the console command:

```
D IOS,MIDAW
```

If MIDAW support is enabled, you can query your FDRPAS source and target DASD volumes to verify that they both support MIDAWs (or not) with the console command:

```
D M=DEV(uuuu)
```

If there is a mismatch in the MIDAW support between your source and target DASD volumes, you must disable MIDAW support globally in order to do the swaps. Use the console command:

```
SETIOS MIDAW=NO
```

DUPLEX COPY

If an FDRPAS source volume is the primary volume in a PPRC, Global Mirror for z/Series (XRC), SRDF, or Dual Copy session, you may leave the session active during the swap. However, you must be aware that after the swap completes the secondary volume is no longer updated. FDRPAS warns you if the source volume in a swap is also the primary volume of a duplex copy (currently this works only if the source volume is in a PPRC session). If you need the duplex copy after the swap, and the new device is capable, you must re-establish the session.

Normally you do not want to establish a duplex copy of the target device before the swap is complete. Since FDRPAS must copy all of the data from the source volume to the target device, all of those writes to the target need to be mirrored on the duplex device and slow down the FDRPAS copy a great deal. If the duplex copy of the data is critical (such as for disaster/recovery), you can establish the duplex copy before the swap as long as you consider the performance implications.

If the source volume is defined to IBM HyperSwap or EMC AutoSwap, there are special considerations that need to be reviewed. See Section 320.6 "FDRPAS and IBM GDPS/PPRC HyperSwap", Section 320.7 "FDRPAS and IBM Basic HyperSwap", and Section 320.8 "FDRPAS and EMC AutoSwap" for details. For additional information on Global Mirror for z/Series (XRC) and FDRPAS, please contact INNOVATION Technical Support.

CONCURRENT COPY (CC)

If FDRPAS detects that a Concurrent Copy (CC) session is active and doing I/O on a source volume at the end of a swap, it delays completing the swap until no Concurrent Copy (CC) I/O has been detected for two minutes. However, this cannot guarantee that the Concurrent Copy (CC) session completes successfully. Because a CC session may involve multiple volumes, it is possible that no CC I/O is done to one of the volumes in the session for many minutes while other volumes are being processed. FDRPAS does not detect the usage of Concurrent Copy (CC) on a source volume unless Concurrent Copy (CC) I/O is detected on that volume.

If a "dormant" Concurrent Copy (CC) session is still active on a source volume when the swap completes, the Concurrent Copy (CC) job fails since the session cannot be transferred to the new device.

CACHE FAST WRITE (CFW)

Cache Fast Write (CFW) is a feature of all cached DASD subsystems that allows data to be held only in cache instead of being written to DASD unless necessary. It is commonly used for sort work areas and may also be used for CICS temporary storage. Although FDRPAS successfully copies the data tracks that were written using Cache Fast Write (CFW), CFW uses a subsystem-wide ID to protect against the lost of CFW data due to the re-initialization of the subsystem. After an FDRPAS swap, the CFW ID of the new subsystem may be different and any application using CFW across the swap may fail. However, new CFW data sets opened after the swap work correctly. CFW is a consideration only for an FDRPAS SWAP, not a SWAPDUMP.

If FDRPAS detects that Cache Fast Write (CFW) is in use on a source volume, it waits until no CFW commands have been issued for two minutes before allowing the swap to complete. In most cases, this avoids CFW problems.

If you prefer, the IDCAMS command SETCACHE can be used to enable and disable CFW for all DASD volumes in a source subsystem before a swap. You may also be able to update global options in your SORT product to disable the use of CFW while you are doing FDRPAS swaps.

In some cases, your SORT product may be able to recover from a Cache Fast Write (CFW) error and complete the sort successfully. Consult your SORT documentation.

MODEL204 from CCA can optionally use Cache Fast Write (CFW) for files on the CCATEMP, CCASERV, and CCASER*xx* DD statements. MODEL204 can fail if it is using CFW on a volume swapped by FDRPAS. This is controlled by the MODEL204 startup parameter CACHE; the default is X'00' (no CFW) and CCA does not recommend using CFW. However, if you have a value other than X'00' for CACHE and want to swap volumes containing those MODEL204 data sets, consult the MODEL204 documentation for information on disabling the use of CFW.

An exposure exists in ADABAS V813 and earlier to report Cache Fast Write (CFW) IO errors after an FDRPAS swap. Contact INNOVATION DATA PROCESSING referencing ISYS incident number **R3021** or SoftwareAG technical support referencing SAGSIS incident number **308599** for details on available maintenance or problem circumvention. There is no exposure to Cache Fast Write (CFW) IO errors in ADABAS V814 and later versions of ADABAS.

ALTERNATE TRACKS

Alternate tracks were used on “real” 3380 and 3390 DASD volumes to recover from defects on the DASD surface. When a track was discovered to be defective, an alternate was assigned from a pool of alternate tracks to take its place. These alternate tracks (1 or 3 cylinders, depending on model) were included in the size of the volume recorded in the VTOC and VTOC index.

Modern DASD, emulating 3380 and 3390 volumes on RAID DASD volumes, do not have traditional alternate tracks. However, some DASD subsystems, especially non-IBM DASD, emulate that pool of alternate tracks even though they are never assigned. On the other hand, IBM DASD subsystems, including the 2105, 2107, DS6000, DS8000, and RAMAC 1, 2, and 3 (but not the RAMAC Virtual Array - RVA) appear to have no alternates at all so their total size appears to be smaller by 1 or 3 cylinders.

This discrepancy has caused problems when volumes are moved from DASD volumes that have or emulate alternate tracks to DASD volumes that have no alternates, and IBM has had to make changes in the VTOC format to accommodate this. A new function was added to ICKDSF (REFORMAT REFVTOC) to make the proper VTOC changes after such a volume move; however, REFVTOC requires that the volume be offline to all but one system during the operation.

FDRPAS automatically makes the proper adjustments in the VTOC and VTOC index, as well as all in-storage tables, on all systems, when a volume is swapped from a device with alternates to one with no alternates, or vice versa. **It is not necessary to run an ICKDSF REFVTOC function after an FDRPAS swap.**

P/390, R/390, FLEX-ES, AND IS/390 INTERNAL DASD

You cannot use FDRPAS to swap volumes on a IBM P/390, R/390, or Flex-ES system. These systems run OS/390 in conjunction with an Intel (P/390 and Flex-ES) or RS/6000 (R/390) system and emulate internal S/390 DASD volumes on files of the host operating system. They do not emulate a control unit that can be used by FDRPAS.

An IBM Integrated Server/390 (IS/390) also runs OS/390 in conjunction with an Intel system, but it supports both emulated internal DASD volumes and external ESCON-attached DASD. FDRPAS cannot be used to swap to or from the internal IS/390 DASD volumes, but it can be used to swap between external DASD volumes.

MP/3000 INTERNAL DASD

You can swap to and from the internal DASD volumes in an IBM MP/3000 system. FDRPAS supports swapping between two internal DASD volumes, or to or from an internal DASD volume to an external channel-attached DASD volume. However, an IBM fix to the internal DASD emulation code may be required; your microcode should be at E26792 level 042 or above.

SYSTEM Z HIGH PERFORMANCE FICON (ZHPF)

There are no customer considerations for the support of System z High Performance FICON (zHPF). To determine if zHPF is enabled for your devices, issue the `D M=DEV(`*`uuuu`*`)` command to see if "ZHPF" is listed on the "FUNCTIONS ENABLED" line. To determine if zHPF is enabled for your channels, issue the "D M=CHP(*xx*)" command to see if "ZHPF" is listed on the "FACILITIES SUPPORTED" line.

3390-54 TO EAV CONSIDERATION

Consider using FDRMOVE with the `EATTR=OPT` option instead of FDRPAS when moving a nearly full 3390-54 to an Extended Address Volume (EAV) since FDRPAS may not yield the desired results. FDRMOVE with the EATTR=OPT option allocates eligible data sets in the cylinder-managed space, while FDRPAS, when moving a nearly full 3390-54, occupies almost all of the track-managed space on the EAV after the move. See "Extended Address Volumes (EAVs)" in Section 325.11 and "EATTR=OPT Considerations" in Section 325.11 for additional information.

FLASHCOPY, EMCSNAP, SHADOWIMAGE, PPRC

FDRPAS target volumes (`SWAPUNIT=`*`uuuu`*) cannot be used by any replication technique (for example, FlashCopy, EMCSNAP, HDS ShadowImage, PPRC). Additionally, you should not have a FlashCopy pool pointing to volumes that will be swapped into.

320.2 FDRPAS SPECIAL SOFTWARE CONSIDERATIONS

This section documents special software considerations for the use of FDRPAS. It is as complete as possible and contains all the considerations that are known to INNOVATION DATA PROCESSING at the time of publication. However, there may be other considerations that have not been identified or that were discovered after publication.

This section should be reviewed carefully before performing any FDRPAS operations.

For the latest updates to software considerations, go to the INNOVATION DATA PROCESSING web site at: http://fdr.com. Click the "FTP Login" link, then type your site access code to login to the registered FTP site.

REQUIRED IBM AND ISV MAINTENANCE

Depending on the level of your operating system, you may need to apply certain IBM PTFs to successfully swap volumes and to avoid problems after the swap. Some of the PTFs are critical; if they apply to your system, they must be applied to avoid problems. Others are recommended; you must decide if the problems that the PTFs fix impact your system.

It may also be necessary to apply maintenance to certain ISV (third-party) software products so that they successfully support FDRPAS swaps. Details are below.

The FTP site referenced above contains a document with the title of *FDRPAS_IBM_and_ThirdParty_Maintenance.txt* that lists all such IBM and other fixes that INNOVATION DATA PROCESSING is aware of. This document is frequently updated, so be sure and get the latest copy before you begin any swaps. This document shows which IBM APARs apply to each level of the operating system, which ones are critical or recommended, and gives a brief description of each.

You must review this list to determine which APARs you must apply. Even some of the critical ones may not apply to your installation. Complete descriptions of the APARs and copies of the fixing PTFs can be obtained from IBM.

PAGE AND SWAP DATA SETS

Volumes containing "*active" local page* or *swap* data sets cannot be swapped with FDRPAS. Volumes with "inactive" page and swap data sets can be swapped.

Rather than swapping volumes with active page and swap data sets, we suggest that you define new page and swap data sets on new volumes on the desired DASD hardware, and migrate the paging activity to them with the console `PAGEDEL REPLACE` command, or the console `PAGEADD` and `PAGEDEL DELETE` commands.

Volumes containing active PLPA and common page data sets (but no active local page data sets) can be swapped with FDRPAS. If the common page data set is updated by a page-out during the swap, it is recopied during the next pass.

JES SPOOL AND CHECKPOINT VOLUMES

JES2 and JES3 spool volumes can be swapped with FDRPAS.

With FDRPAS 5.4/75 and higher and z/OS 1.7 and higher on **ALL** LPARs in the sysplex, FDRPAS can identify JES SPOOL and CHECKPOINT volumes and ensures that they are processed with no other volumes. When **ANY** LPAR has a z/OS system lower than z/OS 1.7, coding the `JESVOL=YES` operand on the MOUNT statement informs FDRPAS that this volume is a JES SPOOL or CHECKPOINT volume and processes this volume with no other volumes. The operand `PRINT=ALL` should not be specified when swapping JES volumes, to avoid potential interlocks.

If a volume to be swapped contains a JES2 checkpoint data set, there is one consideration: if this is a single-system JES2 checkpoint (not MAS - multi-access spool), the default for the HOLD operand on the MASDEF statement in the JES2 startup parameters is `HOLD=9999999`, which causes JES2 to hold a permanent RESERVE on the checkpoint volume. FDRPAS cannot swap a volume while a RESERVE is held, so the swap fails (no harm is done, but the swap is not successful). To circumvent this permanent RESERVE, issue this console command on the system that owns the checkpoint volume to set the RESERVE time to 1 second:

```
$T MASDEF,HOLD=100
```

After the swap you can reissue the command with HOLD=9999999 if you like.

CAUTION:

SWAPDUMP Only!
INNOVATION recommends that users NOT include JES SPOOL and CHECKPOINT volumes in SWAPDUMP and MONITOR jobs along with other volumes; and recommend that JES SPOOL and CHECKPOINT volumes be put in individual jobs (each volume by itself) with no other volumes, using a separate MONITOR job for each JES SPOOL and CHECKPOINT volume, and running these jobs after all the other SWAPDUMP jobs are complete.

SYSPLEX COUPLING DATA SET

A volume containing an active Sysplex Coupling Data Set can be swapped with FDRPAS.

With FDRPAS 5.4/75 and higher and z/OS 1.7 and higher on **ALL** LPARs in the sysplex, FDRPAS can identify active sysplex Coupling Data Set volumes and ensures that they are processed with no other volumes. When **ANY** LPAR has a z/OS system lower than z/OS 1.7, coding the JESVOL=YES operand on the MOUNT statement informs FDRPAS that this volume requires special processing and processes this volume with no other volumes.

CAUTION:

SWAPDUMP Only!
INNOVATION recommends that users NOT include Sysplex Coupling Facility volumes in SWAPDUMP and MONITOR jobs along with other volumes; and recommend that Sysplex Coupling Facility volumes be put in individual jobs (each volume by itself) with no other volumes, using a separate MONITOR job for each Sysplex Coupling Facility volume, and running these jobs after all the other SWAPDUMP jobs are complete.

JES3 MANAGED VOLUMES

FDRPAS supports swapping DASD volumes managed by JES3. JES3-managed DASD volumes are those that are referenced by a DEVICE statement in the JES3 initialization statements (the "INISH deck"). DASD that are not referenced by a JES3 DEVICE statement are managed only by z/OS. Both kinds of DASD can be swapped with FDRPAS on a JES3 system.

If the target device for a SWAP is JES3-managed, then before the SWAP starts, the target device should be offline to JES3 as well as to z/OS. If the device is online, you can use the *VARY or *V command of JES3 to VARY the device offline to both JES3 and z/OS.

JES3 is aware of the swap of JES3-managed DASD volumes and handles them properly. However, you may need to update the DEVICE statements in the INISH deck before the next IPL so that it properly recognizes the new devices.

Note that if you use varying values for the XTYPE= parameter on the JES3 DEVICE statement, the first sub-parameter of XTYPE must match in the source and target devices. For example, a source volume defined as:

```
DEVICE,XTYPE=(DSYSTEM,DA,PR),XUNIT=(C5B,*ALL,S2,ON),NUMDEV=4
```

Can only be swapped to a target DASD volume that also specifies an XTYPE with DSYSTEM.

CICS JOURNAL DATA SETS

There is a consideration for sequential CICS journal data sets. Sequential CICS journal data sets are the old-format journal files that are used in older levels of CICS. **This does not apply to system logger files (including DASDONLY), which is the only supported format for CICS journals in CICS/TS 1.1 and above.**

CICS journal files have a DSORG of PS or PSU and usually have a data set name containing an index level starting with DFHJ. CICS preformats these files so that it can recognize a journal file that was not properly closed. However, when swapping a journal file that is currently inactive (CICS not running), FDRPAS may not copy all of the preformatted tracks, resulting in CICS startup errors.

This problem only occurs for inactive journal files with DSORG=PS, not PSU. DSORG=PSU journal files and all journal files that are currently allocated by a CICS system are copied correctly; only DSORG=PS journal files for inactive CICS systems may have a problem when the CICS system is next restarted.

If you think you might be subject to this consideration, contact INNOVATION DATA PROCESSING for a circumvention.

SYSTEM RESIDENCE VOLUMES

There are two volumes that are used during a system IPL, referenced by device address. These volumes can be moved with FDRPAS, but it is your responsibility to update your IPL parameters and system documentation with the new device addresses before the next IPL. Failure to do so may result in the IPL process using the old devices, with unpredictable results.

One of these is the system residence (IPL) volume, or SYSRES. The address of the SYSRES is specified on your hardware console and is usually called the LOAD ADDRESS.

The other is the IODF volume. The IODF volume contains the I/O configuration data sets and may contain system parameter libraries used during IPL. The address of the IODF volume is also specified on your hardware console as part of a string usually called the LOAD PARAMETER.

Depending on the type of hardware you are using, the LOAD ADDRESS and LOAD PARAMETER may be stored as part of an activation profile. Be sure to update all appropriate activation profiles with the new device addresses.

FDRPAS identifies all swapped volumes with IPL text on the label track or an IODF data set in the VTOC and generates message FDR252 on the console to warn that such parameter updates may be required.

Although FDRPAS modifies the original device of a swapped volume so that it cannot be mounted, this does not prevent an apparently successful IPL from those old devices. IBM has created APARs **OW53222** and OW52127 to address this but if your system does not include those fixes, it is possible to erroneously IPL from the old devices. These fixes are included in 1.5 and above.

SADMP:

If volumes that contain dump data sets used by the IBM IPLable stand-alone system dump program (SADMP) are swapped, you may need to re-generate the SADMP program; check the IBM *z/OS MVS Diagnosis: Tools and Service Aids (GA22-7589)* manual.

ORACLE STORAGETEK PRODUCTS

If you have tape software from Oracle StorageTek prior to HSC 6.2, you must shutdown HSC (Host Software Component) before swapping the volumes containing the HSC CDS (control data set) and restart it after the swap.

If you have HSC 6.2 or above, you must **NOT** shut down HSC on z/OS while swapping the volumes containing the HSC CDS. If HSC is running when a CDS volume is swapped, HSC recognizes that a SWAP has taken place and dynamically updates the CDS with the new device address. If HSC is not running when a CDS volume is swapped, then when you try to bring HSC back up, it detects that the device address of the CDS has changed and refuses to initialize.

On z/VM, HSC in any release needs to be shut down during a SWAP of the volumes containing the CDS.

CA TECHNOLOGIES PRODUCTS

There are considerations if certain products from CA Technologies are in use in your installation.

CA ACF2

If you use CA ACF2 with the SYNCFILE performance enhancer (caches recent database accesses) option enabled and swap a volume that contains the ACF2 SYNCFILE, you need to disable the SYNCFILE process on every system where it is active before performing the swap. There is no function loss while the SYNCFILE process is disabled. To disables the ACF2 SYNCFILE, modify the ACF2 options file to SYNCOPTS NOACTIVATE and refresh the options with the console command: F ACF2,REFRESH(OPTS),SYSID(*cpuid*).

After the swap of the volume completes, change the option back to SYNCOPTS ACTIVATE and refresh the options again.

For additional information, refer to CA Technologies document **TEC477001**.

CA ASTEX

If you use CA ASTEX, you must contact CA Technologies to get any maintenance that affects FDRPAS (or search for FDRPAS on their support site) and apply it. If you do not have all such maintenance applied, you must stop CA ASTEX before swapping any DASD volumes and restart it after swaps are complete.

CA MIM

If you use CA MIM with a DASDONLY control file, you cannot swap the volume containing the currently active control file. One solution is to issue a CA MIM command to switch to the alternate control file while you swap the volume containing the primary control file. Other CA MIM control file options, such as CTCONLY and CTCDASD, should not be a problem.

CA SCHEDULER

If you use CA SCHEDULER at a level **less than** V9.0 and you swap any volume containing data sets used by CA SCHEDULER, you must stop CA SCHEDULER before the swap and restart it after the swap. In V9.0 and above, CA SCHEDULER does not have problems with swap.

SOFTWARE AG ADABAS

There is no exposure to Cache Fast Write (CFW) IO errors in ADABAS V814 and later versions of ADABAS. An exposure exists in ADABAS V813 and earlier to report Cache Fast Write (CFW) IO errors after an FDRPAS swap. Contact INNOVATION DATA PROCESSING referencing ISYS incident number **R3021** or SoftwareAG technical support referencing SAGSIS incident number **308599** for details for any available maintenance or problem circumvention.

ADABAS builds a channel program containing a prefix command, but did not set the flag tilling IOS to add its own prefix. The symptoms of in FDRPAS are swap failures with messages `FDR234** SWAP ERROR REASON=C` and `REASON=E`. Refer to Software AG incident number **1051648** for the recommended fixes for this problem for ADABAS V822 through V825.

ENF SIGNALS

Immediately after an FDRPAS swap completes, an ENF (Event Notification Facility) signal is issued on each system to indicate that the swap was done. Event code 10 (SWAP) is issued, but an ENF exit translates this to event code 28 (SWAP DYNAMIC) on most systems. Software systems that are sensitive to DASD volumes being swapped to new devices listen for those ENF signals, and can take appropriate action to access the volume on its new device address.

Users of the Allocation Control Center (ACC) or Space Recovery System (SRS) products from DTS Software should ensure that fix **DTS22560**, to monitor ENF swap signals, is installed.

If you have other software products that may be sensitive to the device address of a given volume, ask the vendor if they honor ENF SWAP signals.

Some system monitoring products (such as TMON) may not properly report on swapped volumes, if they have not implemented the ENF support. It is usually sufficient to stop and restart those products after the swaps to recognize the new device addresses.

PROGRAMS THAT ACCESS OFFLINE DASD

Avoid executing programs that access offline DASD devices, since they may access or modify an FDRPAS target device during the swap, with unknown results.

ICKDSF can be used to initialize or modify offline DASD volumes. You should not run ICKDSF against an FDRPAS target device. FDRPAS does check to see if the target volume has been re-initialized and terminates the swap.

The IXFP program (used with IBM RVA DASD) and the SVAA program (used with Oracle StorageTek SVA DASD) may access offline DASD when the "space utilization" report is run. This may result in a warning message (SIB0355W) if an FDRPAS target DASD volume is accessed during a swap. No harm is done, and you can either ignore the message or avoid running that report during a swap. Oracle StorageTek fix **L2P005N** for SVAA and IBM fix **L170862** for IXFP resolve this problem by recognizing FDRPAS target devices.

FDRPAS PROGRAM LIBRARY

You can successfully swap the DASD volume that contains the FDRPAS program library. However, INNOVATION DATA PROCESSING recommends that you swap this volume by itself, with no other swaps running.

SYSTEM COUPLE DATA SETS

System Couple Data Sets (CDS) are used in a Parallel Sysplex, in conjunction with a coupling facility. There is a consideration when swapping the volume that contains the *active sysplex CDS*. Other types of CDSs are not affected. The console command:

```
D XCF,COUPLE,TYPE=SYSPLEX
```

can be used to display the primary and alternate sysplex CDS names, and their volsers. Note that after a swap, the device address displayed by this command may still reflect the source volume; this is not a problem. Cross-system Communication Facility (XCF) is sensitive to I/O delays on the sysplex CDS, such as the delays caused when FDRPAS suspends I/O to the volume during a swap.

However, CDS errors are very unlikely to occur. No error is detected unless the XCF "failure detection interval" (default 25 seconds) is exceeded. It is very unlikely that FDRPAS would suspend I/O for that long. Even if the failure detection interval should be exceeded, it results in console message IXC426D. The operator must simply reply "R" to retry and continue.

If you wish to be certain that no problems can occur when swapping a volume that contains the active sysplex CDS, you have several options:

3. You can increase the failure detection interval on every system with the console command:

   ```
   SETXCF COUPLE,INTERVAL=nnn
   ```

4. You can switch to the alternate sysplex CDS with the console command:

   ```
   SETXCF COUPLE,PSWITCH
   ```

   Then you can swap the volume that contains the now-inactive primary sysplex CDS. Afterwards, you can switch back to the primary and swap the volume that contains the alternate.

ACTIVE DATA SETS

Normally, FDRPAS identifies active data sets by testing to see if another task holds a SYSDSN enqueue on the data set. Active data sets are handled with complete integrity during the swap.

For inactive (non-enqueued) Physical Sequential (PS), Partitioned Organization (PO), and VSAM data sets, FDRPAS improves performance by copying only the used tracks within those data sets.

In rare cases, a task may use a data set without holding a SYSDSN enqueue on it. One such case is a started task whose program is in the Program Properties Table (PPT) with the NODSI option (very few programs use this option). If such a task is **updating** a data set without holding the enqueue, FDRPAS may not be able to ensure integrity on the data set. Additional validation is done on PS and VSAM data sets to avoid this problem, but updated Partitioned Organization (PO) data sets may not be detected. If you think you may have this exposure, contact INNOVATION DATA PROCESSING for assistance.

Although JES2 has the NODSI option in its PPT entry, JES2 does not update any of the PDSs that are allocated to it, so this is not an exposure. TSO users or batch jobs that update JES2 PROCLIBs and other PDSs enqueue the data set during the update.

It is not necessary to close any open data sets on volumes being swapped. This includes data sets such as catalogs and databases. The FDRPAS swap is transparent to all applications that use the DASD volume.

SYSRES ALLOCATION BY FDRPAS

You may notice that FDRPAS may do a dynamic allocation to your system residence volume during its operation. This dynamic allocation is normal. This dynamic allocation does not mean that FDRPAS is swapping your SYSRES (unless you have requested FDRPAS to swap your SYSRES volume).

In addition, if an FDRPAS step has an error, the FDR998 or FDR997 message issued by FDRPAS at the end of the step may specify "VOL=*sysres*" with the serial number of your system residence volume. This does not indicate that any error occurred on that volume and can be ignored unless other error messages indicate a true problem with that volume.

CATALOGS THAT USE ENHANCED CATALOG SHARING (ECS)

If a volume that contains an ICF catalog that is enabled for Enhanced Catalog Sharing (ECS) is swapped, ECS sharing is disabled on that catalog. You receive message:

```
IEC378I catname REMOVED FROM ECS DUE TO DDR SWAP
```

on each system. ECS uses a coupling facility to transmit catalog information between systems, so it is only available in a parallel sysplex.

IBM implemented this behavior in APAR OW48166 and fixed it to operate correctly in APAR OA10139, because Enhanced Catalog Sharing (ECS) uses the device address of the catalog in its sharing logic. The text of APAR OW48166 says, in part:

> "The code has been changed to recognize when the volume has been moved to a new hardware device. Any catalogs currently in ECS that are on the affected device are removed from ECS and marked temporarily ineligible. In order for the catalogs to be re-enabled for ECS usage, the installation must issue either the:
>
> ```
> MODIFY CATALOG,ECSHR(ENABLE,catname)
> ```
>
> or
>
> ```
> MODIFY CATALOG,ECSHR(ENABLEALL)
> ```
>
> command. The command to re-enable the catalog(s) may be issued from any system, but should NOT be issued until all systems that share the catalog have removed it from the CF. This may be verified by issuing `MODIFY CATALOG,ECSHR(STATUS)` on all sharing systems. All systems that display the catalogs named in the **IEC378I** message(s) shown above should indicate a status of 'Inact(NonECSAcc)'. Once all sharing systems indicate this, the catalog may be re-enabled for ECS use as described above."

However, it may be safer to remove catalogs from Enhanced Catalog Sharing (ECS) before swapping them, and re-enable Enhanced Catalog Sharing (ECS) after the swap.

ESOTERIC NAMES

Esoteric names are symbolic unit names that are defined in your I/O configuration and relate to specific device addresses. Esoteric names are used in `UNIT=` parameters in JCL and dynamic allocation. For example, `UNIT=SYSDA` is an esoteric name.

If you are swapping a volume that is included in an esoteric name, and the target device is not included in that esoteric name, then any job or dynamic allocation that uses the esoteric name to allocate the volume fails after the swap is complete. You must either update the esoteric name to include both the source and target devices before the swap, or update the esoteric name immediately after the swap. Consult IBM documentation for information on defining and changing esoteric names.

ALLOCATION BY SPECIFIC DEVICE ADDRESS

It is possible to use specific unit addresses in `UNIT=` JCL parameters and dynamic allocations to allocate specific DASD volumes, for example, `UNIT=3A2` or `UNIT=/125A`.

It is rare that JCL uses specific unit addresses, but it is more likely that programs that dynamically allocate DASD volumes might use z/OS services to get the unit address of a DASD volume and use that address in a dynamic allocation. If a job or dynamic allocation uses a specific unit address obtained before an FDRPAS swap completes, but does the allocation after the swap, it fails.

JCL using specific unit addresses should be changed (to use generic or esoteric names, preferably) and programs using dynamic allocation may need to be rerun.

ENQUEUE PROPAGATION

FDRPAS does enqueues with major names of FDRPAS, FDRPASQ, and FDRPASU* with SCOPE=SYSTEMS to indicate that swaps are in progress. The enqueues are used to detect duplicate swap requests and inhibit certain operations. It is required that these enqueues be propagated to all systems involved in the swap. If these enqueues are not propagated to some systems, FDRPAS is not be able to detect duplicate swap requests and the ISPF panels on systems running MONITOR tasks do not detect the swap in progress until synchronization has completed on all systems. However, FDRPAS still operates correctly even if all the systems involved are not part of the same GRS complex or MIM complex. You should not convert the FDRPAS, FDRPASQ, and FDRPASU* enqueues to SCOPE=SYSTEM. CA MIM users may need to add these major names to a MIM QNAME list in order to propagate them.

FDRPAS USE OF ICKDSF

If you are using FDRPAS to swap a volume to a larger device, such as a 3390-9 to a 3390-27, you must specify LARGERSIZE=OK. At the end of the swap, if the volume has an active indexed VTOC (VTOCIX), FDRPAS invokes ICKDSF to rebuild the VTOCIX (BUILDIX) to reflect the new size of the volume. FDRPAS coordinates the VTOCIX update on multiple systems.

WARNING: ***The VTOC or VTOCIX size on the volume may not be adequate after you swap it to a larger DASD volume and start adding new data sets to the volume. If you are licensed for FDRMOVE, you can use the EXPANDVTOC operand to expand the VTOC on these volumes. See Section 310.13 "SWAPBUILDIX and EXPANDVTOC Statement" for further information on EXPANDVTOC.***

- A 3390-27 requires a minimum of (2) tracks for the Indexed VTOC.
- A 3390-54 and a 3390-A EAV require a minimum of (4) tracks for the Indexed VTOC.
- If the Indexed VTOC is not allocated with the minimum specification, then a U0888 ABEND and the following FDR254 error message results when FDRPAS invokes ICKDSF BUILDIX IXVTOC:

```
FDR254 ICKDSF MSG=ICK516I uuuu I/O ERROR DETECTED DURING
VTOC CONVERSION: ERROR CODE=5
```

IBM has determined that an enqueue interlock can occur when doing this type of BUILDIX if the SYSVTOC and SYSZVVDS resources are being treated differently by your cross-system enqueue facility (GRS or CA MIM). For GRS, IBM added a requirement to the *GRS Planning Guide* stating that SYSZVVDS and SYSVTOC must either both be in the RESERVE CONVERSION RNL or both be in the SYSTEMS EXCLUSION RNL. For CA MIM, the equivalent rules must be in effect. Failure to do this may cause the BUILDIX to hang.

STATIC AND DYNAMIC UCBS

In your I/O configuration, defined with HCD, the UCB for each DASD device can be defined as "installation static" or "dynamic". Consult the IBM HCD documentation for details.

FDRPAS can swap a volume from a static UCB to a dynamic UCB and vice versa. If you have never before had dynamic UCBs in your installation, you should verify that locally-written and vendor (ISV) programs include support for dynamic UCBs. Programs using the IBM UCBSCAN and UCBLOOK services must include the operand DYNAMIC=YES to find dynamic UCBs.

4-DIGIT DEVICE ADDRESSES AND UCBS ABOVE THE LINE

FDRPAS can swap between DASD devices with 3- and 4-digit device addresses and between UCBs that are located below the 16MB line and above the 16MB line (LOCANY=YES in the HCD configuration).

However, before you swap a volume to a 4-digit device or a device with its UCBs above the line, you should be sure that all software using the volume has been upgraded to support such devices. It is possible that the volume was on a 3-digit device or a UCB below the line precisely because the software using it has not yet been upgraded.

NOTE: When FDRPAS swaps between a UCB below the line and one above the line, the target device UCB is below the line. However, after the next IPL, it reverts to an above the line UCB.

FULL-VOLUME RESTORE AND COPY

If FDR, DFSMSdss, or another DASD backup/restore product is used to do a full-volume restore or copy to a volume that FDRPAS is currently swapping to another device, you should examine the volume after the swap is complete to ensure that the device characteristics in the VTOC and VTOC index (VTOCIX) are correct.

A full-volume restore or copy operation may make changes to the volume size in the VTOC and VTOCIX when:

- The target device is larger than the volume on the backup (for a restore) or the source volume (for a copy).
- The target device has no alternate tracks while the original volume does have alternate tracks, or vice versa (see "Alternate Tracks" in Section 320.1).

Unfortunately, the full-volume restore/copy program may make decisions about the VTOC changes to make based on the characteristics of the volume at the time the restore/copy begins. When FDRPAS is swapping the device, this would be the original source volume. However, the restore/copy may not complete until after FDRPAS has swapped the volume to its new device. The new device may be a larger device than the original. The decisions made by the restore program before the swap may not be valid after the swap, so the changes it makes to the VTOC and VTOCIX may not be valid.

In addition, FDRPAS itself may make changes to the VTOC and VTOCIX when the alternate tracks and device size of the target device are different from the source volume. FDRPAS and the restore program may make conflicting changes to the VTOC. Even worse, if the restore/copy program changes the location of the VTOC or VTOCIX during the restore, FDRPAS may update the wrong copy of the VTOC or VTOCIX.

If you know that a full-volume restore or copy was done during an FDRPAS swap, you should use tools such as FDREPORT, COMPAKTOR, IEHLIST, or other DASD mapping software to validate that the number of data cylinders in the VTOC and VTOCIX is correct.

In any case, it makes little sense to use FDRPAS to swap a volume if you are going to completely replace it with a restore or copy. If you know that a restore/copy is done, it is simpler to restore or copy the volume to its new device directly instead of using FDRPAS at all.

NOTE: This consideration does not apply to data set restores and copies.

SYSTEM NAMES

Many FDRPAS messages, and other parts of this document, refer to "systems" or "system names". Some FDRPAS messages refer to them as CPUs.

These system names come from the field CVTSNAME in the Communication Vector Table (CVT) of each system (sometimes referred to as a "system image", an "image" of the operating system). The system name is assigned by the IEASYS*xx* member of PARMLIB. Each system involved in an FDRPAS swap must have a unique system name. To display the name of a system, type this console command on a console connected to the system:

```
D GRS
```

and you receive a display similar to:

D GRS

```
ISG343I 12.46.18 GRS STATUS 348
SYSTEM    STATE       COMM    SYSTEM    STATE       COMM
CPUB      ACTIVE              CPUC      ACTIVE       YES
```

The first system listed (CPUB in this example) is the system name of this system.

CPU SERIAL NUMBERS

Some FDRPAS messages include CPU serial numbers. You may also need to specify a CPU serial number on an EXCLUDE statement.

The CPUID value is the 10-character CPU serial number of a system image. When you run a SIMSWAP job, the CPUID values display on the FDR233 message as shown in this example:

```
FDR303  CARD IMAGE --   SIMSWAP  TYPE=FULL
FDR303  CARD IMAGE --   MOUNT    VOL=HI17C2,SWAPUNIT=17C1
FDR233  CPU WITH (SERIAL# 0212342818) IS ATTACHED TO VOL=HI17C2            - HTC 2107900 TO HTC 2107900
FDR233  CPU WITH (SERIAL# 0112342818) IS ATTACHED TO VOL=HI17C2            - HTC 2107900 TO HTC 2107900
FDR233  CPU WITH (SERIAL# 0512342818) IS ATTACHED TO VOL=HI17C2            - HTC 2107900 TO HTC 2107900
FDR233  CPU WITH (SERIAL# 0912342818) IS ATTACHED TO VOL=HI17C2            - HTC 2107900 TO HTC 2107900
FDRW66  SWAP OF VOL=HI17C2 TO UNIT=17C1 NEEDS TO BE STARTED ON   4 SYSTEMS
```

You can also get the CPUID value is the on a specific z/OS system by executing this console command from a console attached to that system:

```
D M=CPU
```

You get a response similar to:

D M=CPU

```
IEE174I 15.34.53 DISPLAY M 899
PROCESSOR STATUS
ID  CPU                 SERIAL
00  +                    0212342818
01  +                    1212342818
```

Note that the first digit may be non-zero if you have a multi-processor system, as shown in this example. The first digit is always zero in FDRPAS message and parameters. The second digit is an LPAR number, if you have a system with multiple LPARs defined.

However, on a z990 system (last four digits are 2064) or any successor system, the first two digits may be the LPAR number, since those systems support more than 15 LPARs.

DYNAMIC MONITORING (DYNMON)

In a GRS or MIM complex environment, FDRPAS MONITOR tasks can run without specifying any devices or MOUNT statements, letting the MONITOR tasks dynamically allocate all the volumes.

320.3 FDRPAS Z/VM AND LINUX FOR SYSTEM Z CONSIDERATIONS

If you run z/VM or Linux for System z in your installation, there may be special considerations. The following outlines the considerations, but if you are uncertain, **please call INNOVATION DATA PROCESSING for guidance.**

FDRPAS runs only on z/OS systems, either a native z/OS system or a z/OS guest running on a virtual machine under z/VM.

FDRPAS can be used to swap volumes used by z/OS guest systems, either full-volume dedicated DASD volumes, z/VM mini-disks, or Linux for System z volumes. It is also possible to swap z/VM-owned volumes or Linux for System z volumes using a native z/OS system if the z/VM systems and Linux for System z systems that use it are shut down. All of these variations are discussed in the rest of this section.

FDRPAS identifies systems that participate in a swap by their CPU IDs (also called "serial numbers"). A z/VM system has such a CPU ID determined by the hardware that the z/VM system is running on, but z/OS guest systems running on z/VM virtual machines have simulated CPU IDs controlled by z/VM; these simulated CPU IDs may be the same as the z/VM hardware ID (the default) or may be overridden in the z/VM directory (CPUID parameter). The way that your installation sets the CPU IDs of z/OS guest systems affects how they impact FDRPAS.

There are several different ways that you might be running z/VM, Linux for System z, and z/OS guest systems:

z/VM with No z/OS Guests

If you run native z/OS systems along with z/VM systems that have no z/OS guest systems running beneath then, then the z/VM systems do not have to participate in the FDRPAS SWAP operation at all. All FDRPAS SWAP functions are executed on the native z/OS systems. However, if the source volume in an FDRPAS operation is accessible to a z/VM system, the DASD hardware may report the CPU ID of the z/VM system; in this case you should use the EXCLUDE statement (see Section 310.4 "SWAP Task EXCLUDE Statement") to exclude it. If the source volume is in an IBM 3990-3 or other control unit that does not report the CPU IDs, then simply specify the #SYSTEMS= parameter with a value that does not include the z/VM system, only the z/OS systems with access to the volume.

z/VM with a Single z/OS Guest and Dedicated DASD

If the FDRPAS source DASD is dedicated (via the DEDICATE statement in the z/VM directory or the z/VM ATTACH command) to one z/OS guest (even if you have other z/OS guests active), then you do not need to do anything special. Simply start an FDRPAS SWAP task or MONITOR task on the z/OS guest as you would on any other z/OS system. The target device must also be defined to z/VM and dedicated to the z/OS guest.

If the DASD is attached to other native z/OS systems (not under z/VM), run FDRPAS tasks on those systems as well.

z/VM Full-Volume Mini-Disks

If the FDRPAS source DASD is defined as a full-volume z/VM mini-disk (via the MDISK statement in the directory entry of one or more z/OS guest systems), there are special procedures. These notes apply even if only one z/OS guest is using the volume, but full-volume mini-disks are used mainly when multiple z/OS guests must share the volume. The target device must also be defined to z/VM and defined as a full-volume mini-disk to every z/OS guest.

A full-volume z/VM mini-disk does not support some of the special commands that are used by FDRPAS, so a normal MONITOR task does not work. You need to use a special MONITOR task with the following control statements:

```
MONITOR TYPE=JOIN
MOUNT SWAPUNIT=xxxx
```

Where "*xxxx*" specifies a single specific target unit address (not a prefix). To swap multiple volumes, you must submit separate MONITOR tasks. **Do not** submit this type of MONITOR task on the system where the SWAP task runs.

Z/VM WITH MULTIPLE Z/OS GUESTS

On the SWAP statement, in the SWAP task, you must specify the parameter:

```
#SYSTEMS=nnn
```

Where "*nnn*" is the actual number of z/OS systems (including all the z/OS guests) that participate in the SWAP. It is very important that you specify this value accurately.

Every z/OS system that participates in a SWAP must have a unique CPU ID. If you have multiple z/OS guests running under z/VM, and two or more of them are involved in a SWAP, by default z/VM lets each guest use the same CPU ID and FDRPAS cannot tell them apart. There are two ways to address z/VM guests with the same CPU ID:

5. You can give each z/OS guest a unique virtual CPU ID by using the CPUID parameter in the z/VM directory entry for each guest. The CPU ID values assigned can be any valid value, as long as they are unique.
6. You can tell FDRPAS to use a simulated CPU ID in each MONITOR tasks by inserting this statement after the MONITOR statement in the FDRPAS input:

   ```
   PROFILE  CPUID=xxxxxxxxxx
   ```

   and providing any valid 10-character CPU ID, as long as it is unique for each guest. This value is used only by FDRPAS and does not affect any other use of the CPU ID. If you include a PROFILE statement. then, in addition, you must specify `MAXCARDS=3` on the `MONITOR TYPE=JOIN` statement.

NOTE: The CPU IDs must be unique in the last nine characters. INNOVATION DATA PROCESSING recommends that you make the CPUID unique by changing only the second digit (normally used as the LPAR number) so that the rest of the serial still matches the hardware CPU ID of the system.

An FDRPAS SWAP job stream for execution on a z/OS guest with full-volume mini-disks that is shared by two other guests might look like:

```
//SWAP     EXEC PGM=FDRPAS,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  SWAP     TYPE=FULL,#SYSTEMS=3
  MOUNT    VOL=ABC123,SWAPUNIT=125A
/*
```

The FDRPAS MONITOR task for execution on the other two z/OS guests might look like:

```
//MONITOR  EXEC PGM=FDRPAS,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  MONITOR  TYPE=JOIN,MAXCARDS=3
  PROFILE  CPUID=0512342818 (if needed to provide unique CPU ID)
  MOUNT    SWAPUNIT=125A
/*
```

On some types of DASD hardware, FDRPAS generates message FDR234 REASON=L because the CPU ID of the z/OS guest the SWAP task is executing on is not in the list of CPU IDs returned by the DASD hardware. This occurs because the DASD hardware only knows the CPU IDs of the physical systems and are unaware of the virtual CPU IDs that are used by the z/OS guest systems. The FDR234 REASON=L message is normal and can be ignored as long as the proper number of systems participate. If one of the z/OS guests uses the real CPU ID of the processor hardware (the z/VM default), you can run the SWAP task on that system to avoid the FDR234 REASON=L message.

SWAPPING Z/VM-OWNED VOLUMES

It is possible to use FDRPAS to swap z/VM-owned volumes, including z/VM system volumes and volumes containing z/VM mini-disks. The SWAP must be done from a native z/OS system (not a z/OS guest running under z/VM), and **all of the z/VM systems that use the volumes must be shut down**. FDRPAS has no MONITOR that can run on z/VM, so it has no way of tracking updates by a z/VM system or any way to swap the volume on an active z/VM system. The source volume and the target device must be accessible by the native z/OS system, and by the z/VM systems that use it. After the SWAP, when you re-IPL the z/VM systems, they find the volume on its new device.

It is possible to move inactive z/VM volumes with other programs, such as FDR full-volume copy and similar utilities, but FDRPAS is a very convenient way of moving them, so the details are documented here.

z/VM-owned volumes are normally initialized with the CPVOLUME function of ICKDSF (or the z/VM command CPFMTXA that invokes ICKDSF). This builds a “dummy” VTOC on the volume, that starts and ends on Cylinder 0 Head 0. FDRPAS recognizes this dummy VTOC, knows that it is a z/VM volume, and copies all tracks on the volume (since it cannot tell what tracks are really in use). Message FDR023 is issued to indicate that the dummy VTOC was found.

Therefore, for z/VM-owned volumes that were properly initialized by CPVOLUME, you can run a normal FDRPAS SWAP. If the z/VM volume is accessible by multiple native z/OS systems, then you must run FDRPAS MONITOR tasks as well.

If the z/VM systems were shutdown but not deactivated (LPAR deactivate or native system power-off), then the DASD hardware may still report the CPU ID of the z/VM system. You may need to add `EXCLUDE CPUID=` statements to the FDRPAS SWAP task to avoid errors.

WARNING:

If the z/VM-owned volume was initialized with the ICKDSF INIT command or some other function that builds a normal VTOC on the volume instead of the z/VM “dummy” VTOC, then FDRPAS is not able to recognize the volume as a z/VM-owned volume. Instead, FDRPAS copies only the tracks indicated in that VTOC, usually just the VTOC itself. If the SWAP of a z/VM volume does not produce the FDR023 message or copies less than all tracks on the volume, this is probably the cause. The volume may not be usable and mini-disk contents may be lost.

The above can also occur if the z/VM volume contains several z/OS-formatted mini-disks, where the first such mini-disk starts on cylinder 0. FDRPAS sees only the VTOC on that first mini-disk and copies tracks based only on its contents, so the additional z/OS mini-disks are lost.

You can swap z/VM volumes without a dummy VTOC by adding a SELECT statement after all MOUNT statements in the SWAP task input as in this example:

```
MOUNT  VOL=VM4201,SWAPUNIT=6201
MOUNT  VOL=VM4202,SWAPUNIT=6202
MOUNT  VOL=VM4203,SWAPUNIT=6203
SELECT FROM(CYL=0),TO(CYL=65519)
```

The ending cylinder value must be as large as or larger than the highest cylinder number on the largest DASD to be swapped; 65519 as shown handles any DASD up to the size of a 3390-54. This SELECT statement causes all tracks on all swapped volumes to be copied, so it can be specified when swapping any z/VM-owned volumes, whether they have dummy VTOCs or not. However, if the SWAP task also includes some non-z/VM volumes, it unnecessarily copies all tracks on those volumes as well, so INNOVATION DATA PROCESSING recommends swapping only z/VM volumes in a step with the SELECT statement.

If you like, you can identify z/VM-owned volumes that do not have a dummy VTOC using this job stream:

```
//STEP01   EXEC PGM=FDREPORT,REGION=0M
//SYSPRINT DD SYSOUT=*
//ABRMAP   DD SYSOUT=*
```

```
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  XSELECT  DSN=****VTOC,EXTENTS=00000000,VOL=vol  or VOLG=pfx
  REPORT   FIELD=(SDSN,VOL,SIZE,EXTENTS)
  PRINT    SORT=COMBINE
/*
//STEP02   EXEC PGM=FDREPORT,REGION=0M
//SYSPRINT DD SYSOUT=*
//ABRMAP   DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  XSELECT  VOLG=pfx
  PRINT    DATATYPE=VOLDATA
/*
```

For a z/VM volume with a dummy VTOC, STEP01 prints a normal report line, while a volume with a normal VTOC prints:

```
FDR495   NO DATA SETS MATCHED SELECTION CRITERIA -- VOLSER=serial
```

You need to use the SELECT statement above for the latter volumes if the volumes are z/VM-owned.

You should also verify that the report from STEP02 is reporting the correct sizes for the z/VM volumes listed. The size that is reported in STEP02 is the size that FDRPAS sees and uses to move the contents of the volume, so this is important for z/VM volumes.

NOTE: The CHECKSOURCE=YES operand should not be specified with processing z/VM volumes.

SWAPPING LINUX ON SYSTEM Z-OWNED VOLUMES

It is possible to use FDRPAS to swap Linux for System z volumes. The SWAP must be done from a native z/OS system (not a z/OS guest running under z/VM), and **all of the Linux for System z systems that use the volumes must be shut down**. The source volume and the target device must be accessible by the native z/OS system, and by the Linux for System z systems that use it. After the SWAP, when you re-IPL the Linux for System z systems, they find the volume on its new device.

It is possible to move inactive Linux for System z volumes with other programs, such as FDR full-volume copy and similar utilities, but FDRPAS is a very convenient way of moving them, so the details are documented here.

Linux for System z-owned volumes are normally initialized with the CPVOLUME function of ICKDSF (or the Linux for System z command CPFMTXA that invokes ICKDSF). This builds a "dummy" VTOC on the volume, that starts and ends on Cylinder 0 Head 0. FDRPAS recognizes this dummy VTOC, knows that it is a Linux for System z volume, and copies all tracks on the volume (since it cannot tell what tracks are really in use). Message FDR023 is issued to indicate that the dummy VTOC was found.

Therefore, for Linux for System z-owned volumes that were properly initialized by CPVOLUME, you can run a normal FDRPAS SWAP. If the Linux for System z volume is accessible by multiple native z/OS systems, then you must run FDRPAS MONITOR tasks as well.

If the Linux for System z systems were shutdown but not deactivated (LPAR deactivate or native system power-off), then the DASD hardware may still report the CPU ID of the Linux for System z system. You may need to add `EXCLUDE CPUID=` statements to the FDRPAS SWAP task to avoid errors.

WARNING:

If the Linux for System z-owned volume was initialized with the ICKDSF INIT command or some other function that builds a normal VTOC on the volume instead of the Linux for System z "dummy" VTOC, then FDRPAS is not able to recognize the volume as a Linux for System z-owned volume. Instead, FDRPAS copies only the tracks indicated in that VTOC, usually just the VTOC itself. If the SWAP of a Linux for System z volume does not produce the FDR023 message or copies less than all tracks on the volume, this is probably the cause. The volume may not be usable and mini-disk contents may be lost.

The above can also occur if the Linux for System z volume contains several z/OS-formatted Linux-disks, where the first such Linux-disk starts on cylinder 0. FDRPAS sees only the VTOC on that first Linux-disk and copies tracks based only on its contents, so the additional z/OS Linux-disks are lost.

You can swap Linux for System z volumes without a dummy VTOC by adding a SELECT statement after all MOUNT statements in the SWAP task input as in this example:

```
MOUNT  VOL=LX4201,SWAPUNIT=6201
MOUNT  VOL=LX4202,SWAPUNIT=6202
MOUNT  VOL=LX4203,SWAPUNIT=6203
SELECT FROM(CYL=0),TO(CYL=65519)
```

The ending cylinder value must be as large as or larger than the highest cylinder number on the largest DASD to be swapped; 65519 as shown handles any DASD up to the size of a 3390-54. This SELECT statement causes all tracks on all swapped volumes to be copied, so it can be specified when swapping any Linux for System z-owned volumes, whether they have dummy VTOCs or not. However, if the SWAP task also includes some non-Linux for System z volumes, it unnecessarily copies all tracks on those volumes as well, so INNOVATION DATA PROCESSING recommends swapping only Linux for System z volumes in a step with the SELECT statement.

If you like, you can identify Linux for System z-owned volumes that do not have a dummy VTOC using this job stream:

```
//STEP01   EXEC PGM=FDREPORT,REGION=0M
```

```
//SYSPRINT DD SYSOUT=*
//ABRMAP   DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  XSELECT  DSN=****VTOC,EXTENTS=00000000,VOL=vol  or VOLG=pfx
  REPORT   FIELD=(SDSN,VOL,SIZE,EXTENTS)
  PRINT    SORT=COMBINE
/*
//STEP02   EXEC PGM=FDREPORT,REGION=0M
//SYSPRINT DD SYSOUT=*
//ABRMAP   DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  XSELECT  VOLG=pfx
  PRINT    DATATYPE=VOLDATA
/*
```

For a Linux for System z volume with a dummy VTOC, STEP01 prints a normal report line, while a volume with a normal VTOC prints:

```
FDR495   NO DATA SETS MATCHED SELECTION CRITERIA -- VOLSER=serial
```

You need to use the SELECT statement above for the latter volumes if the volumes are Linux for System z-owned.

You should also verify that the report from STEP02 is reporting the correct sizes for the Linux for System z volumes listed. The size that is reported in STEP02 is the size that FDRPAS sees and uses to move the contents of the volume, so this is important for Linux for System z volumes.

NOTE: The `CHECKSOURCE=YES` operand should not be specified with processing Linux for System z volumes.

320.4 FDRPAS E-MAIL NOTIFICATION FACILITY

FDRPAS includes a facility that can send an e-mail message when an FDRPAS operation on a volume completes either unsuccessfully or successfully. This can be useful when you, the FDRPAS user, are not on-site or are not monitoring FDRPAS. You can also send a message to a pager or cell phone if you have an alpha-numeric pager or cell phone capable of receiving text messages.

REQUIREMENTS FOR SUCCESSFUL E-MAIL

To send an e-mail from FDRPAS, you must:

- Be running the IBM TCP/IP product (a standard component of z/OS).
- Have an external internet connection to your mainframe that allows you to send e-mail or communicate with an external e-mail server. If you have a firewall it must allow the e-mail or server connection.
- Have access to a mail server that supports Simple Mail Transfer Protocol (SMTP). This can be the optional SMTP mail server that is delivered with the IBM TCP/IP product (running on your z/OS system), or an external mail server (check with your mail server administrator to see if it supports SMTP).
- Run the FDRPAS task under a RACF user id with an OMVS segment (or the equivalent in other security systems), since the IBM TCP/IP product uses UNIX System Services (USS) sockets.

The "Test E-mail Parameters Example" in Section 320.4 can be used to test if the above requirements have been met and your e-mail parameters are correct.

SENDING E-MAIL

By default, FDRPAS sends an e-mail only when certain triggering error messages are issued. Optionally, you can send e-mail indicating a successful operation as well. If FDRPAS issues consecutive triggering messages with identical text, the messages after the first are ignored.

You can specify the subject line for the e-mail and you can provide any message text that you like, with any number of lines of text. FDRPAS supports substituting certain strings in the subject or message text, which allows the message to include the FDRPAS job and/or step name, the name of the system on which the operation executed, and the date/time of the e-mail. The text of the triggering FDRPAS message is included, which usually identifies the volume.

FDRPAS can also e-mail the complete set of messages that are generated by the FDRPAS operation on the volume. This is supported only if you are using the FDRPAS subtask option `MAXTASKS=`*`nn`*.

The e-mail can be sent to up to five recipients. If your mail server supports group names, you can send it to a group name that can be distributed to many recipients.

If your pager or cell phone company supports sending text messages by e-mail, the message can be directed to your pager or cell phone. Consult your provider for the proper e-mail address and requirements.

FDREMAIL DD STATEMENT

To invoke the FDRPAS e-mail facility for a given FDRPAS step or started task, include in the step JCL an FDREMAIL DD statement that points to the e-mail control statements, which are described below. This DD statement can specify a sequential data set, a member of a PDS, or in-line control statements. If it is a data set, it must be LRECL=80, RECFM=F or FB. If the FDREMAIL DD statement specifies DD DUMMY, it is ignored. It is also ignored in MONITOR tasks except for control statement errors.

No additional options are required, the presence of the DD statement is sufficient to activate the facility. However, if you want to be notified of successful FDRPAS operations as well as failures, you must specify the operand EMSG=OK on the SWAP, SWAPDUMP, or SWAPBUILDIX statement.

Examples:

```
//FDREMAIL DD DISP=SHR,DSN=yourid.FDRPAS.EMAIL
//FDREMAIL DD DISP=SHR,DSN=yourid.FDRPAS.CNTL(EMAIL)
//FDREMAIL DD *
 …  e-mail statements …
```

The PASPROC procedure distributed with FDRPAS, used for starting FDRPAS as a started task as well as in batch jobs, includes an FDREMAIL DD statement that defaults to DSN=NULLFILE (equivalent to DD DUMMY). You can specify the parameter `EMAIL=dsname` to activate the e-mail facility, for example,

```
S PASPROC.SWAP1,PARM=' SWAP TYPE=FULL/ MOUNT VOL=ABC123,SWAPUNIT=4321',
EMAIL=FDRPAS.EMAIL
```

SYSTCPD DD STATEMENT

The SYSTCPD DD statement points to TCP/IP parameters used on your system. It is optional, but if it is present, and the SYSTCPD DD statement contains the IP address of a domain name server, then you can specify the name of a mail server instead of the IP address; TCP/IP uses the domain name server to resolve the name. Your installation may have defined the name of a default TCPDATA data set, in that case the DD statement can be omitted.

If it is omitted and your installation has not defined a default TCPDATA, you need to specify the IP address of the mail server in “dotted-decimal” format.

If you do not know what to put on this DD statement, consult the system programmer responsible for TCP/IP on your system.

Example:

```
//SYSTCPD  DD DISP=SHR,DSN=TCPIP.TCPDATA(CPUC)
```

BASIC E-MAIL STATEMENTS

The FDREMAIL DD statement must point to the e-mail statements that define your mail server, the “sender's” e-mail address, the recipients' e-mail addresses, and the e-mail text. See “Enhanced E-Mail Statements” in Section 320.4 for details on sending multiple messages and customizing the message.

These statements must appear in the order shown below, although optional statements can be omitted. Only the first 72 characters of each line are processed. Lines with an asterisk (*) in column1 are treated as comments, except when they are data lines.

TCPNAME

This optional statement contains the name of the TCP/IP address space (started task) on the system where you are executing. If omitted, FDRPAS connects to the first active TCP/IP address space, so it can usually be omitted unless you need to use a particular TCP/IP stack. There must be exactly one space after TCPNAME. For example,

```
TCPNAME ENSRV001
```

MAILSERVER

This optional statement contains the name or IP address of your mail server. This server must support the Simple Mail Transfer Protocol (SMTP) (consult the mail server administrator if you are not sure). If omitted, FDR attempts to use the SMTP server running on your z/OS system; if that SMTP server is not active, you must provide a MAILSERVER statement. There must be exactly one space after MAILSERVER.

If you have included a SYSTCPD DD statement or your installation has defined a default TCPDATA data set, and the parameters specify the address of a domain name server, you can specify the name of the mail server. For example,

```
MAILSERVER MAIL.MYCOMPANY.COM
```

You can also specify the IP address of the mail server, in standard "dotted-decimal" format. If a name server is not available, you must specify the IP address. For example,

```
MAILSERVER 123.45.6.234
```

NOTE: You can use the TCP/IP PING command under TSO to get the IP address of your mail server, for example, `PING MAIL.MYCOMPANY.COM`

FROM:

This required statement defines the e-mail address of the "sender" of the e-mail, in any format that is accepted by your mail server, up to 66 characters. The `FROM:` e-mail address is used by Simple Mail Transfer Protocol (SMTP) for notification of undeliverable messages. It may also be used by the mail server for authentication, so the `FROM:` e-mail address may need to be a valid address known to that server. For example,

```
FROM:<STORMGMT@MYCOMPANY.COM>
```

TO:

CC:

This statement defines the e-mail address of a recipient of the e-mail, in any format that is accepted by your mail server, up to 66 characters. You must include at least one `TO:` statement and can have up to five `TO:` or `CC:` statements. Some examples are:

```
TO:<JANEDOE@MYCOMPANY.COM>
TO:John_Smith@mycompany.com
CC:Bob Jones<BJONES@MYCOMPANY.COM>
```

SUBJECT:

This required statement defines the subject for the e-mail message. The text may contain symbolic substitutions as described below. For example,

```
SUBJECT: FDRPAS FAILURE JOB=&JOBNAME
```

Data Lines

Optionally include any number of data lines after the `SUBJECT:` statement to define the e-mail message to be sent. Do not forget that pagers and cell phones may truncate long messages. The text may contain symbolic substitutions as described below. For example,

```
FDRPAS JOB &JOBNAME FAILED ON SYSTEM &SYSTEMS
&MESSAGE
```

SYMBOLIC SUBSTITUTION

The e-mail subject line or data lines may contain certain symbolic parameters, which FDRPAS replaces with appropriate values before the e-mail is sent. The values are:

&JOBNAME

The name of the FDRPAS job or started task.

&STEPNAME

The name of the FDRPAS step.

&SYSTEMS

The name of the system on that FDRPAS was executing.

&EMAILDATE

The date that the e-mail was generated by FDRPAS, in mm/dd/yyyy format.

&EMAILEDAT

The date that the e-mail was generated by FDRPAS, in dd/mm/yyyy ("European") format.

&EMAILTIME

The time that the e-mail was generated by FDRPAS, in hh:mm:ss.t format.

&MESSAGE

&MSG

The text of the FDRPAS message that triggered the e-mail. This symbolic must be the last or only thing on the line where it appears. If this symbolic does not appear in the e-mail text, the message is automatically included as the last line of the message, so it is usually not required unless you want to include the message in the subject text or position it.

&ALLMSG

Results in multiple lines in the e-mail, containing all of the messages associated with the volume. `&ALLMSG` only works if FDRPAS subtasking (`MAXTASKS=`*nn*) was used. This symbolic must be the only thing on the line where it appears. Do not use on a `SUBJECT:` line.

ENHANCED E-MAIL STATEMENTS

In order to meet more complex notification requirements, the FDRPAS e-mail facility can also:

- Send varying e-mail text to different recipients (for example, send a one line notification to one address, and send the full set of messages to another).
- Send different e-mail texts depending whether the volume completed successfully or unsuccessfully. You can even vary the e-mail text depending on the FDRPAS message that triggered the e-mail.

The enhanced e-mail is invoked by an additional e-mail statement:

EMAIL:

Indicate the start of an e-mail message definition and terminates the data lines of the previous message. If omitted, there is only a single e-mail definition (see "Basic E-Mail Statements" in Section 320.4). The `EMAIL:` statement can optionally appear before the first `FROM:` statement and it is inserted after the data lines of a message definition to separate the next message definition. It has an optional operand `MSG=` that is separated from `EMAIL:` by one space.

The `EMAIL:` statement may be followed by the `TCPNAME`, `MAILSERVER` and/or `FROM:` statements, but if omitted, the previous values are used. It must be followed by one to five `TO:` or `CC:` statements to define the recipients of this e-mail. The `TO:` and `CC:` statement(s) are followed by a new `SUBJECT:` statement and optionally new data lines to define the e-mail text.

The `MSG=` operand controls when the following e-mail is sent and has several forms:

MSG=ERR – Send only for unsuccessful FDRPAS volume operations (when the triggering message is followed by `**`).

MSG=OK – Send only for successful FDRPAS volume operations (when the triggering message is not followed by `**`). Note that you must specify the EMSG=OK operand on the SWAP, SWAPDUMP, or SWAPBUILDIX statement to invoke the e-mail processor for successful completions.

MSG=*nnn* – MSG=(*nnn*,*nnn*,...) – Send only if the message that triggered the e-mail is FDR*nnn*. The triggering messages currently include FDR999 (successful, only if EMSG=OK or EMSG=999 was specified), FDR997 and FDR998 (completed with errors), FDR319 (subtask ABEND), FDR302 (control card error), and certain FDR234 (SWAP error) messages.

An example of enhanced e-mail statements is:

```
MAILSERVER 12.132.14.178
FROM:JOHNDOE@MYCOMPANY.COM
EMAIL: MSG=ERR
*  SEND ALL FAILURE MESSAGES TO DICK
TO:DICK<DICK@MYCOMPANY.COM>
SUBJECT: FDRPAS SWAP FAILED &JOBNAME &STEPNAME &SYSTEMS
EMAIL: MSG=302
*  IF JOB FAILED FOR CONTROL CARD ERRORS, NOTIFY JANE
TO:JANE<JANE@MYCOMPANY.COM>
SUBJECT: FDRPAS CONTROL CARD ERROR
&MESSAGE
JOB &JOBNAME MUST BE RE-SUBMITTED
EMAIL: MSG=OK
*  FOR SUCCESSFUL SWAP, SEND ALL MESSAGES TO SPOT
TO:SPOT<SPOT@MYCOMPANY.COM>
SUBJECT: FDRPAS MESSAGES: &MESSAGE
&ALLMSG
```

SEND AN E-MAIL IF SWAP FAILS EXAMPLE

Send an e-mail if the swap of volume DATA22 fails. Since no SYSTCPD DD statement is provided, the MAILSERVER statement must specify the IP address of the mail server. The e-mail includes the text of the error message.

```
//SWAP     EXEC PGM=FDRPAS,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//FDREMAIL DD *
MAILSERVER 12.132.14.178
FROM:<ABC@MYCOMPANY.COM>
TO:<XYZ@MYCOMPANY.COM>
SUBJECT: DATA22 SWAP FAILED
/*
//SYSIN    DD *
  SWAP     TYPE=FULL
  MOUNT    VOL=DATA22,SWAPUNIT=07C3
/*
```

SEND AN E-MAIL FOR EACH STEP INDICATING SUCCESS OR FAILURE EXAMPLE

Send an e-mail for each volume in this step that indicates success or failure. Since a SYSTCPD DD statement is provided, the `MAILSERVER` statement can specify the name of the mail server. “PASLIST” is the name of a group that is interpreted by the mail server. The e-mail includes the text of the message to identify the volume.

```
//SWAP     EXEC PGM=FDRPAS,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTCPD  DD DISP=SHR,DSN=TCPIP.DATA(SYSTEM1)
//FDREMAIL DD DISP=SHR,DSN=STGMGMT.FDRPAS.CNTL(EMAIL)
//SYSIN    DD *
  SWAP     TYPE=FULL,EMSG=OK
  MOUNT    VOL=TSO001,SWAPUNIT=1A**
  MOUNT    VOL=TSO002,SWAPUNIT=1A**
  …
  MOUNT    VOL=TSO099,SWAPUNIT=1A**
/*
```

Data set `STGMGMT.FDRPAS.CNTL(EMAIL)` contains:

```
MAILSERVER MAIL.MYCOMPANY.COM
FROM:JOHN DOE<JOHNDOE@MYCOMPANY.COM>
TO:PASLIST@MYCOMPANY.COM
SUBJECT: FDRPAS SWAP ENDED &SYSTEMS, JOB &JOBNAME, STEP &STEPNAME
MESSAGE - &MESSAGE
```

SEND AN E-MAIL IF FDRMOVE JOB FAILS EXAMPLE

Send an e-mail if the FDRMOVE job fails. Since no SYSTCPD DD statement is provided, the MAILSERVER statement must specify the IP address of the mail server. The e-mail includes the text of the error message.

```
//MOVE     EXEC PGM=FDRMOVE,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//FDREMAIL DD *
MAILSERVER 12.132.14.178
FROM:<ABC@MYCOMPANY.COM>
TO:<XYZ@MYCOMPANY.COM>
SUBJECT: FDRMOVE JOB FAILED
/*
//SYSIN    DD *
  FASTMOVE TYPE=DSF
  SELECT   CATDSN=PAYROLL.ACCOUNTS,NVOL=SY3100
/*
```

SEND AN E-MAIL TO SEVERAL RECIPIENTS IF STEP FAILS EXAMPLE

Send an e-mail to several recipients for each volume in this step that fails. Since a SYSTCPD DD statement is provided, the MAILSERVER statement can specify the name of the mail server. The e-mail includes all messages generated for the failing volume. Also, send a 1-line notification to a pager.

```
//SWAP     EXEC PGM=FDRPAS,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTCPD  DD DISP=SHR,DSN=TCPIP.DATA(SYSTEM1)
//FDREMAIL DD DISP=SHR,DSN=STGMGMT.FDRPAS.CNTL(EMAILALL)
//SYSIN    DD *
  SWAP     TYPE=FULL,MAXTASKS=10
  MOUNT    VOL=TSO001,SWAPUNIT=1A**
  MOUNT    VOL=TSO002,SWAPUNIT=1A**
  …
  MOUNT    VOL=TSO099,SWAPUNIT=1A**
/*
```

Data set STGMGMT.FDRPAS.CNTL(EMAILALL) contains:

```
MAILSERVER MAIL.MYCOMPANY.COM
FROM:<DILBERT@MYCOMPANY.COM>
TO:<DOGBERT@MYCOMPANY.COM>
TO:<CATBERT@MYCOMPANY.COM>
CC:<DILBERT@MYCOMPANY.COM>
SUBJECT: SWAP FAILED ON &SYSTEMS &EMAILDATE &EMAILTIME &MSG
&ALLMSG
EMAIL:
TO:Pager<9735552345@VTEXT.COM>
SUBJECT: FDRPAS ERROR &MSG
```

TEST E-MAIL PARAMETERS EXAMPLE

This job stream can be used to test your e-mail parameters and verify that you can successfully send an e-mail via SMTP. This job stream causes a control statement error that invokes the e-mail facility.

```
//BADSWAP  EXEC PGM=FDRPAS,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//FDREMAIL DD *
MAILSERVER 12.132.14.178
FROM:Me<ME@MYCOMPANY.COM>
TO:Me<ME@MYCOMPANY.COM>
SUBJECT: E-MAIL TEST
/*
//SYSIN    DD *
  SWAP     TYPE=ZZZZ   DELIBERATE CONTROL STATEMENT ERROR
/*
```

320.5 FDR EXTENDED MCS SOFTWARE CONSOLE (FDREMCS)

FDREMCS is a program written by INNOVATION DATA PROCESSING to provide the ability to issue z/OS console commands from a step that uses an Extended MCS (EMCS) software console This program also has some additional special functions.

FDREMCS can issue any z/OS console command (subject to RACF controls).

To execute FDREMCS, the following JCL statements are required:

EXEC STATEMENT The EXEC statement specifies the program name (PGM=FDREMCS). A REGION= parameter is not required.

STEPLIB DD STATEMENT Since FDRPAS is normally installed in a library separate from any other FDR products; a STEPLIB DD statement is usually required, pointing to that library.

SYSPRINT DD STATEMENT Specifies the output message data set. It must be present and is usually a SYSOUT data set, but it may be assigned to DASD or tape.

SYSUDUMP DD STATEMENT Specifies the ABEND dump data set. Usually specifies a SYSOUT data set. Although the SYSUDUMP DD statement is not required, we strongly urge you to always include this DD statement, so that we can help you diagnose error conditions. If you have a debugging aid product on your system that would prevent the desired dump, please add the appropriate one of these statements to the JCL so that a fully-formatted dump is produced.

```
//ABNLIGNR DD DUMMY    Turn off Abend-Aid
//CAOESTOP DD DUMMY    Turn off CA OPT II & CA SYMDUMP
//DMBENAN  DD DUMMY    Turn off DumpMaster
//ESPYIBM  DD DUMMY    Turn off Eye-Spy
//IDIOFF   DD DUMMY    Turn off IBM Fault Analyzer
//PSPOFF   DD DUMMY    Turn off Softworks Performance Essential
```

SYSUT1 DD STATEMENT Specifies a data set to be copied to the SYSUT2 DD. Usually a sequential data set; could be a member of a PDS. Must have RECFM=F or FB and LRECL=80; may have any blocksize that is a multiple of 80.

SYSUT2 DD STATEMENT Specifies a data set to be copied from the SYSUT1 DD. Usually a sequential data set; could be a member of a PDS. FDREMCS sets RECFM=FB and LRECL=80, and sets BLKSIZE=6160 if the block size is not specified.

SYSIN DD STATEMENT Specifies a data set containing the control statements for FDREMCS. Usually a DD * data set. It is required.

FDREMCS CONTROL STATEMENTS FDREMCS accepts only three control statements: `CONSOLE=`, `TERMINAL=`, and `COMMAND=`. Each of the control statements must start on a new line; you cannot put more than one control statement on a line. Comment lines are indicated by an asterisk in column 1.

CONSOLE=

Specifies the name you want assigned to the console, from one to eight alphanumeric characters. Only one `CONSOLE=` statement may appear.

Default: FDREMCS.

TERMINAL=

Specifies the terminal name assigned to the console, from one to eight alphanumeric characters. Only one `TERMINAL=` statement may appear. This name is logged on the master console but is not used for anything else.

Default: The value specified for `CONSOLE=`.

COMMAND=

Specifies a z/OS console command to be issued through the EMCS console. All text after the equal sign through column 71 is used; no quotes are required. The COMMAND= statement may be continued. A continuation is indicated by the last non-blank character in columns 1 through 71 being a comma (,). The continuation begins with the first non-blank character on the next input line. Multiple continuation lines are allowed, but the total length of the command text must not exceed 120 characters.

More than one COMMAND= statement may be input.

CONSOLE RESPONSES

FDREMCS waits up to five seconds after each command has been issued, and displays any responses directed to the EMCS console in that time. This may include responses to the commands that are issued, but may also include other unrelated messages. Responses returned after the five second limit are not displayed. FDREMCS has no way of knowing if the command you specified was completed successfully or had an error.

SAMPLE FDREMCS JOB

This example shows how to issue a DISPLAY command.

```
//DISPLAY  EXEC PGM=FDREMCS
//STEPLIB  DD DISP=SHR,DSNAME=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  CONSOLE=MYCONS
  COMMAND=DISPLAY SMS
/*
```

The console name MYCONS is optional, but is recommended to be sure that the name does not conflict with other EMCS consoles. As long as the responses to the DISPLAY SMS are returned within five seconds, they are displayed on SYSPRINT.

320.6 FDRPAS AND IBM GDPS/PPRC HYPERSWAP

Geographically Dispersed Parallel Sysplex/Peer-to-Peer Remote Copy (GDPS/PPRC) HyperSwap is an IBM offering that non-disruptively swaps I/O activity from the primary volumes to the secondary volumes of PPRC mirrored pairs, when a problem occurs on any primary volume. Automation is used to swap all mirrored volumes at the same time.

A GDPS/PPRC HyperSwap cannot occur at the same time as an FDRPAS SWAP. FDRPAS cannot SWAP a volume while it is eligible to be swapped by HyperSwap. FDRPAS offers facilities to disable HyperSwap for the minimum amount of time while FDRPAS does its UCB swaps, and then re-enable HyperSwap. With these facilities, HyperSwap remains active while FDRPAS does all of the copying of data from the source volumes to the target volumes. HyperSwap is disabled by using the `HYPERSW DISABLE` command, which is much faster than the older `HYPERSW OFF` command. `HYPERSW DISABLE` just tells GDPS not to do a HyperSwap, after which `HYPERSW ON` tells GDPS that it is okay again to do a HyperSwap when needed; while `HYPERSW OFF` deconstructs the HyperSwap environment, after which `HYPERSW ON` has to rebuild the HyperSwap environment.

This section, Section 320.6 "FDRPAS and IBM GDPS/PPRC HyperSwap", deals with GDPS HyperSwap. See Section 320.7 "FDRPAS and IBM Basic HyperSwap" for information on Basic HyperSwap.

In most cases, the number of volumes under the control of GDPS/PPRC HyperSwap is very large. To do an FDRPAS SWAP of volumes that are being managed by GDPS/PPRC HyperSwap, follow the procedure in Section 305.8 "FDRPAS for Large Scale Synchronized Migration", specifying `CONFIRMSWAP=YES` on the SWAP jobs. In one of the `MONITOR TYPE=CONFIRMSWAP` jobs, include special steps detailed below to disable and re-enable HyperSwap at the appropriate times. In this section, the `MONITOR TYPE=CONFIRMSWAP` job with the special steps is called the "special 4-step job". Run standard `TYPE=CONFIRMSWAP` jobs on the other LPARs running SWAP jobs. With `CONFIRMSWAP=YES`, FDRPAS allows HyperSwap to remain enabled during the data copy phase of each volume and until the SWAP operation is completed by `MONITOR TYPE=CONFIRMSWAP`; otherwise a volume being managed by HyperSwap would not be processed.

TERMINOLOGY

This section uses some terms that have special meanings for GDPS:

Controlling system(s)

One or two LPARs in the GDPS sysplex that manage the GDPS functions. Also known as the "K-system(s)". The controlling system(s) share a minimum of DASD and other resources with the production systems, so that they are not affected by problems that may cause outages on the production systems. If there are two controlling systems, one is the master controlling system and the other is the alternate controlling system.

Production system

Any LPAR in the GDPS sysplex that is not a controlling system. It can be a system that runs a production workload, or a test or development system.

RULES, RECOMMENDATIONS, AND CONSIDERATIONS

The following rules, recommendations, and considerations apply to this procedure.

- Most installations that use GDPS consider it critical to keep the time that GDPS is disabled to an absolute minimum. The procedure in Section 305.8 "FDRPAS for Large Scale Synchronized Migration" fulfills this requirement. With this procedure, HyperSwap is disabled only once during a migration of up to 15,000 volumes.
- In order for full duplexing to be in effect as soon as the FDRPAS SWAP tasks are complete, it is necessary to add the FDRPAS target devices to the GDPS/PPRC configuration before the FDRPAS SWAP tasks start. While FDRPAS is copying the data from the source device to the target device, PPRC copies the data from the target (which is a PPRC primary device) to its PPRC secondary device. The double copy causes the FDRPAS SWAP to run slower than it would otherwise, but it is necessary in order to have the target device synchronized with its PPRC secondary when GDPS is re-enabled after the FDRPAS SWAP.
- MONITOR tasks must be run on the GDPS controlling system(s), since the volumes being swapped are online on these LPARs.
- FDRPAS has a rule that the target devices must be offline on the LPAR running the SWAP task and on all LPARs running MONITOR tasks. In general, GDPS has a rule that all of the PPRC primary volumes must be online in the controlling system(s). In GDPS 3.9 and above, there is an exception that PPRC primary volumes are allowed to be offline in the controlling system(s) if they are marked as Reserve Storage Pool (RSP) volumes. Therefore, the FDRPAS target volumes should be initialized as RSP volumes before starting the SWAP tasks.

  This is different from the usual procedure for FDRPAS, which does not require the target devices to be initialized in any way before the SWAP.

  To initialize volumes as RSP, run the INIT command of ICKDSF with the RESERVED parameter. Do not specify a reserve storage pool name with the `OWERNID` or `RESERVEPOOLNAME` parameters, since the target volumes are not really in a reserve storage pool. The `RESERVED` parameter was added to ICKDSF by PTF **UK70219**, available 2011/08/04, for APAR **PM16856**.

  If you are running GDPS at a lower level than 3.9, or if you do not have the support for the RESERVED parameter in ICKDSF INIT, you must start out with the target volumes as offline but not RSP. This may result in an alert being raised by GDPS monitoring on the controlling system(s).
- If you use the I/O timing facility as a trigger for HyperSwap, then you should disable it while running FDRPAS by issuing the console command:

  ```
  SETIOS MIH,IOTHSWAP=NO
  ```

  At the end of each copy pass, FDRPAS temporarily suspends all I/O to the source volume. Under some conditions, this can cause I/O requests to remain on the queue for longer than the I/O timing limit, especially if you have set a short limit (one minute or less). This condition is not an error and should not be allowed to trigger HyperSwap.

  After you are finished running FDRPAS, you can re-enable I/O timing as a trigger for HyperSwap by issuing the console command:

  ```
  SETIOS MIH,IOTHSWAP=YES
  ```

SWAP AND MONITOR JOBS

The SWAP and MONITOR jobs should be generated with GENSWAP following the procedure in Section 305.8 "FDRPAS for Large Scale Synchronized Migration", with the following considerations:

- The example in Section 305.8 "FDRPAS for Large Scale Synchronized Migration" is for 15,000 volumes, but the setup is the same for any number of volumes.
- The example is set up for SWAP/SWAPDUMP jobs on six LPARs, MONITOR tasks on a total of ten LPARs, and `MAXACTIVESWAPS=10`. These are all just examples. Adjust all these values appropriately for your environment.

- Specify SWAP instead of SWAPDUMP.
- Specify `CONFIRMSWAP=YES` instead of `CONFIRMSPLIT=YES`.
- Do not specify `CPYVOLID=YES`. Instead, if the target volumes are initialized as RSP volumes before the SWAP, specify `LABEL=SWAP` so that after the SWAP, the source devices become RSP volumes and GDPS considers it valid for them to be offline.

Example of SWAP command:

```
SWAP      TYPE=FULL,MAXTASKS=64,CONFIRMSWAP=YES,LABEL=SWAP,
          LARGESWAP=16000,SWAPID=&&&,ALLOWPAV=YES,LARGERSIZE=OK
```

COMPLETING THE SWAP OPERATION

Once **ALL** of the volumes have reached synchronization, the SWAP operation can be completed. "Completed" means disabling HyperSwap, issuing the actual UCB swaps, and re-enabling HyperSwap. On one of the systems running SWAP jobs, run the special 4-step job described below. On any other LPARs running SWAP jobs, run a regular MONITOR TYPE=CONFIRMSWAP job as shown in Section 305.8 "FDRPAS for Large Scale Synchronized Migration". A single GENSWAP run can generate the special 4-step job and all of the regular MONITOR TYPE=CONFIRMSWAP jobs.

SPECIAL 4-STEP JOB

Step 1 (CONFIRM) of the Special 4-Step Job

The first step (1) is the regular MONITOR TYPE=CONFIRMSWAP. This step checks that all of the SWAP tasks are active, to make sure that none of the volumes have been left out or have failed. It then waits for all of the source volumes to reach the "ready to confirm" stage, in order to coordinate among the SWAP tasks and ensure that all of the volumes are swapped at the same time. Finally it confirms all of the SWAP tasks.

Step 2 (DISABLE) of the Special 4-Step Job

The SWAP tasks do not complete as long as the volumes are eligible to be swapped by HyperSwap. The second step (2) of the special job uses program FDREMCS (Section 320.5 "FDR Extended MCS Software Console (FDREMCS)") to issue a `MODIFY (F)` command to NetView to disable HyperSwap. FDREMCS monitors the command responses for 60 seconds, looking for the **GEO551I** message indicating that HyperSwap has been disabled. (This is an exception to the general rule in Section 320.5 "FDR Extended MCS Software Console (FDREMCS)" that FDREMCS monitors responses for only 5 seconds, and that it has no way of knowing if the command specified was completed successfully or had an error.) When this message appears, FDREMCS sets a flag to notify FDRPAS. Meanwhile, the SWAP tasks cycle, testing for this flag every few seconds. As soon as the flag is set, the SWAP tasks complete and terminate.

NOTE 1: Change *netview* to the name of the NetView started task on the LPAR where the job runs.

NOTE 2: NetView parameters may need to be modified to accept z/OS `MODIFY (F)` commands from a console called "FDREMCS" (or whatever you specify for `CONSOLE=`); consult the NetView manuals for details.

NOTE 3: Security rules may need to be modified to allow the EMCS console to issue the MODIFY command; for details, consult the z/OS manuals and the manuals for RACF or your security system.

Step 3 (WAITTERM) of the Special 4-Step Job

The third step (3) of the special job step waits for the SWAP tasks to terminate on all of the selected volumes. A `//*SWAPNEXT` statement precedes this step to tell GENSWAP to generate all of the same MOUNT commands as for the CONFIRM job.

Step 4 (ENABLE) of the Special 4-Step Job

The fourth and final step (4) of the special job uses program FDREMCS to issue a `MODIFY (F)` command to NetView to re-enable HyperSwap.

NOTE: Change *netview* to the name of the NetView started task on the system where the job runs.

4-STEP JOB EXAMPLE

Here is an example of the GENSWAP run to generate the special 4-step job and all of the regular MONITOR TYPE=CONFIRMSWAP jobs to complete the swap operation in a GDPS HyperSwap environment.

- A `//*SWAPNEXT` statement is used before each CONFIRM job to instruct GENSWAP to generate all of the same MOUNT statements for each job.
- An override of the `//PAS.INTRDR DD` statement directs the output to SYSOUT for viewing instead of submitting the output directly to the internal reader for execution. Be sure to review the generated jobstream well in advance of the time to run the actual CONFIRM job. When it is time to run the actual CONFIRM job, remove the `//PAS.INTRDR DD` statement to allow the generated jobs to be submitted to the LPARs for execution.

This example can be found in the JCL library installed with FDRPAS with member name PA32006A.

```
//GENCNFRM EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.INTRDR DD SYSOUT=*
//PAS.SYSIN  DD *
  GENSWAP  TYPE=FULL,MAXGEN=16000,
           SUBMITDELAY=0,MAXCARDS=16000
  MOUNT    VOL=*,SWAPUNIT=5***,CUID=65950
  …
  MOUNT    VOL=*,SWAPUNIT=A***,CUID=22398
/*
//PAS.PASJOB DD DATA,DLM=SS
1 //*SWAPJOB
//PASCA    JOB …
/*ROUTE    XEQ JESCPUA
//CONFIRM  EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN  DD *
  MONITOR  TYPE=CONFIRMSWAP,MAXCARDS=16000,LARGESWAP=16000
  MOUNT    VOL=&&&&&&,SWAPUNIT=&&&&
/*
2 //DISABLE  EXEC PASPROC,PROG=FDREMCS,LIB=fdrpas.loadlib
//PAS.SYSIN  DD *
  COMMAND=MODIFY netview,HYPERSW DISABLE
/*
3 //*SWAPNEXT
//WAITTERM EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN  DD *
  MONITOR  TYPE=WAITSWAPTERM,MAXCARDS=16000
  MOUNT    VOL=&&&&&&,SWAPUNIT=&&&&
/*
4 //ENABLE   EXEC PASPROC,PROG=FDREMCS,LIB=fdrpas.loadlib
//PAS.SYSIN  DD *
  COMMAND=MODIFY netview,HYPERSW ON
/*
//*SWAPNEXT
//PASCB    JOB …
/*ROUTE    XEQ JESCPUB
//CONFIRM  EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN  DD *
  MONITOR  TYPE=CONFIRMSWAP,MAXCARDS=16000,LARGESWAP=16000
  MOUNT    VOL=&&&&&&,SWAPUNIT=&&&&
/*
```

```
//*SWAPNEXT
…    (replicate CONFIRM for all LPARs running main SWAPs)
//*SWAPNEXT
//PASCF    JOB …
/*ROUTE    XEQ JESCPUF
//CONFIRM  EXEC PASPROC,LIB=fdrpas.loadlib
//PAS.SYSIN  DD *
  MONITOR  TYPE=CONFIRMSWAP,MAXCARDS=16000,LARGESWAP=16000
  MOUNT    VOL=&&&&&&,SWAPUNIT=&&&&
/*
SS                    <- End of outer PASJOB statements
```

320.7 FDRPAS AND IBM BASIC HYPERSWAP

Basic HyperSwap is an IBM facility included in z/OS, which automates the swap of a large number of Metro Mirror (PPRC) primary volumes to their secondary devices, in case of a planned or unplanned outage. FDRPAS cannot SWAP a volume while it is eligible to be swapped by HyperSwap. A specialized program, FDRHXBLK, is available to block Basic HyperSwap before FDRPAS does SWAP tasks, and to unblock Basic HyperSwap after the FDRPAS SWAP tasks are complete (block/unblock support requires z/OS 1.12 or above, or z/OS 1.9 through 1.11 with the fix for APAR OA26509). With Block and Unblock, Basic HyperSwap does not need to be brought down to do an FDRPAS SWAP. Replication continues, and volumes remain in PPRC synchronization, but Basic HyperSwap does not initiate a SWAP for the short period when it is blocked. The Block is not in effect while FDRPAS is copying the data from the source volumes to the targets. The Block is only issued after the targets are synchronized and FDRPAS is ready to do the actual SWAP of the UCBs. The Unblock is done as soon as all of the actual SWAP tasks have been completed. The time depends on the number of DASD volumes being swapped and the number of participating LPARs, and may be only a few seconds. Basic HyperSwap is aware of the FDRPAS SWAP tasks and automatically updates its configuration information. Since the period when Basic HyperSwap is blocked is very short, it should not be necessary to do the FDRPAS SWAP tasks of all DASD volumes in the Basic HyperSwap configuration at the same time; the installation is able to do the FDRPAS SWAP tasks in convenient groups, such as 64 or 256 volumes at a time.

If Basic HyperSwap is being blocked, then each monitor task also issues a block. Since a block request only applies to one sysplex, this is necessary in case the LPAR in which the monitor is running is in a different sysplex. If Basic HyperSwap is being blocked, then when a monitor does the UCB SWAP for the last volume in a group, the monitor issues an unblock.

This section (FDRPAS and IBM Basic HyperSwap) deals with Basic HyperSwap. See Section 320.6 "FDRPAS and IBM GDPS/PPRC HyperSwap" for information on GDPS HyperSwap.

To do an FDRPAS SWAP of volumes that are being managed by Basic HyperSwap, specify `CONFIRMSWAP=YES` on the SWAP jobs, and run a special 4-step job (described below) while the SWAP jobs are running, to Block and Unblock Basic HyperSwap at the appropriate times. With `CONFIRMSWAP=YES`, FDRPAS allows HyperSwap to remain enabled during the data copy phase of each volume; otherwise a volume being managed by HyperSwap would not be processed. Since each FDRPAS job can do up to 64 DASD volumes concurrently, multiple FDRPAS SWAP jobs may be required.

RULES The following rules and recommendations apply to this procedure.

- All of the SWAP jobs in a group must run on the same LPAR.
- The special 4-step job must be executed on the same LPAR as the SWAP job(s).
- All of the SWAP jobs in a group must complete before the next group is started.
- All of the DASD volumes in the group must be in the Basic HyperSwap configuration. You cannot SWAP HyperSwap and non-HyperSwap volumes at the same time.
- It is recommended that all of the volumes in a group be the same size[1] (model), for example, all 3390-3 or all 3390-27.
- This procedure does not place a limit on the number of volumes in a group, but we suggest one to four SWAP jobs at a time with up to 64 volumes each.
- The SWAP jobs must use `CONFIRMSWAP=YES`.

1. The reason is that if sizes are mixed, the smaller volumes are copied faster, and their SWAP jobs wait for the larger volumes. If all of the volumes in a group are the same size, there should be less idle time.
   The above applies if the volumes have about the same percent used. What we really mean is that all of the volumes in a group should have a similar amount of data, that is a similar amount of used space.

- MAXACTIVESWAPS=YES is recommended.
- It is a customer responsibility to set up PPRC for the target devices, and to include the target devices in the HyperSwap configuration.
- If you use the I/O timing facility as a trigger for HyperSwap, then you should disable it while running FDRPAS by issuing the console command:

  ```
  SETIOS MIH,IOTHSWAP=NO
  ```

  At the end of each copy pass, FDRPAS temporarily suspends all I/O to the source volume. Under some conditions, this can cause I/O requests to remain on the queue for longer than the I/O timing limit, especially if you have set a short limit (one minute or less). This condition is not an error and should not be allowed to trigger HyperSwap.

  After you are finished running FDRPAS, you can re-enable I/O timing as a trigger for HyperSwap by issuing the console command:

  ```
  SETIOS MIH,IOTHSWAP=YES
  ```

SWAP JOB

Here is an example of the SWAP step:

```
//SWAP     EXEC PGM=FDRPAS,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  SWAP     TYPE=FULL,CONFIRMSWAP=YES,MAXTASKS=64,
           MAXACTIVESWAPS=YES
  MOUNT    VOL=vvvvvv,SWAPUNIT=uuuu
…
/*
```

NOTE: A maximum of 64 MOUNT statements are permitted per job step.

MONITOR TASKS

As usual, FDRPAS MONITOR tasks are required on all other LPARs. Each MONITOR task can handle up to 64 concurrent swaps, so you may need multiple MONITOR tasks. It may be convenient to limit each monitor task to 32 targets, matching the 32 DASD volumes in each SWAP step. See Section 310.23 "MONITOR SWAP Examples" for examples of MONITOR task steps.

CONFIRM, BLOCK, WAIT, & UNBLOCK JOB

When the FDRPAS SWAP jobs are running, submit a special 4-step job.

Step 5 (CONFIRM) of the Confirm, Block, Wait, & Unblock Job

The confirm step waits for all DASD volumes to reach the "ready to confirm" stage, and then it confirms them.

```
//CONFIRM  EXEC PGM=FDRPAS,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  MONITOR  TYPE=CONFIRMSWAP
  MOUNT    VOL=vvvvvv
…
/*
```

NOTE: Add additional MOUNT statements for ALL the volumes to be swapped. If there are more than 250 MOUNT statements, change the MAXCARDS= operand accordingly.

Step 6 (BLOCK) of the Confirm, Block, Wait, & Unblock Job

However, the swaps do not complete as long as the DASD volumes are eligible to be swapped by HyperSwap. The block step executes program FDRHXBLK to block Basic HyperSwap.

```
//BLOCK    EXEC PGM=FDRHXBLK,PARM=BLOCK
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
```

NOTE: FDRHXBLK does not use a SYSIN DD statement or control statements. The DASD volumes are not listed because the blocking of Basic HyperSwap is a global operation that applies to the whole system.

As soon as the SWAP task see that Basic HyperSwap has been blocked, they perform the actual UCB swaps and terminate.

Step 7 (WAITTERM) of the Confirm, Block, Wait, & Unblock Job

The WAITTERM step waits for the SWAP tasks to terminate on all of the selected volumes.

```
//WAITTERM EXEC PGM=FDRPAS,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  MONITOR  TYPE=WAITSWAPTERM
  MOUNT    VOL=vvvvvv
…
/*
```

NOTE: Add additional MOUNT statements for **ALL** the volumes to be swapped. It is very important to have the same list of volumes in this step as in the CONFIRMSWAP step. If there are more than 250 MOUNT statements, change the `MAXCARDS=` operand accordingly.

Step 8 (UNBLOCK) of the Confirm, Block, Wait, & Unblock Job

After all of the SWAP tasks have terminated, the unblock step executes program FDRHXBLK to unblock Basic HyperSwap.

```
//UNBLOCK  EXEC PGM=FDRHXBLK,PARM=UNBLOCK
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
```

NOTE: FDRHXBLK does not use a SYSIN DD statement or control statements. The DASD volumes are not listed because the unblocking of Basic HyperSwap is a global operation that applies to the whole system.

320.8 FDRPAS AND EMC AUTOSWAP

FDREMCS is especially useful when combined with the FDRPAS enhanced support for AutoSwap. FDRPAS can do the work of copying data from old to new devices, but cannot do the final SWAP while AutoSwap is active. The enhanced support allows FDRPAS to wait until AutoSwap has been disabled, then quickly complete the swap of all active volumes. FDREMCS, when used in conjunction with some special FDRPAS steps, is used to issue the EMC AutoSwap disable command, and later the re-enable command, so that AutoSwap disablement time is reduced to a few minutes at most.

More details on this process are contained in member **AUTOSWAP** in the ICL library on the use of the enhanced AutoSwap support.

This example shows how to issue a MODIFY (F) command to disable AutoSwap.

```
//DISABLE  EXEC PGM=FDREMCS
//STEPLIB  DD DISP=SHR,DSNAME=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  COMMAND=MODIFY EMCCGRP,DAS,SETSWAP GROUP groupname DISABLE RUN DETAIL
/*
```

The name of the EMC ConGroup task on your system may not be EMCCGRP so you need to substitute the actual name.

FDRPAS SUPPORT FOR AUTOSWAP

FDRPAS SWAP is initiated for a set of DASD volumes by using the `CONFIRMSWAP=YES` operand. With `CONFIRMSWAP=YES`, FDRPAS allows AutoSwap to remain enabled during the data copy phase of each volume. Since each FDRPAS job can do up to 64 DASD volumes concurrently, multiple FDRPAS SWAP jobs may be required.

Turn Off Multiple FDRPAS SWAP Jobs May Be Required

When the FDRPAS SWAP jobs are running, a special 4-step job is submitted.

1. An FDRPAS CONFIRM step that confirms all the DASD volumes.
2. A step that issues the command necessary to globally disable AutoSwap.
3. A special FDRPAS step that waits for all of the FDRPAS SWAP jobs to complete.
4. A step that issues the command necessary to globally enable AutoSwap.

The process can be repeated for each set of 256 DASD volumes to be swapped, with a short AutoSwap outage for each set. It is possible to do more than 256 DASD volumes in a set, but there are limitations on the total number of active swaps so it is not recommended.

A sample of this 4-step job is in member **AUTOSWAP** in the ICL library as mentioned above.

325 FDRMOVE

325.1 FDRMOVE INTRODUCTION

FDRMOVE builds on the proven technology of FDRPAS (volume migration) and FDRINSTANT (instant data set copy) to provide non-disruptive or minimally-disruptive movement of individual data sets.

KEY FEATURES OF FDRMOVE

The key features of FDRMOVE are:

- Automation of the movement of data sets, using simple JCL and control statements.
- Unattended operation: once started, FDRMOVE can execute for long periods of time without human intervention or monitoring.
- Handling of active data sets: when a data set is identified as active (because a SYSDSN enqueue is held by another task), FDRMOVE begins testing the enqueue on the data set at frequent intervals. As soon as the enqueue becomes free, the data set is moved. Some data sets become naturally inactive, but for others, you may need to take action to free them for brief intervals.
- Use of fast data replication hardware facilities, such as FlashCopy on IBM and HDS DASD, EMCSNAP on EMC DASD, and SNAP on Oracle StorageTek DASD. The FASTMOVE function of FDRMOVE moves source volumes into the target subsystem (with a technique called “transit station”), using FDRPAS, where fast replication can be used to quickly move data sets when they become available.
- FDRMOVE can move all types of data sets including VSAM, DB2, IMS, ADABAS, and non-VSAM data sets. FDRMOVE can move individual pieces of a multi-volume data set, or the entire data set.
- FDRMOVE can select data sets to be moved by volume or SMS storage group, selecting all data sets on a volume, or filtering the data sets with a mask. Data sets can also be selected from the catalog.
- Output volumes can be specified by volume serial number, or by SMS storage group.
- The FDR E-mail notification facility is supported. See Section 320.4 “FDRPAS E-mail Notification Facility” for details and examples.

FDRMOVE can be used to move data at the data set level to new DASD, and is particularly useful for combining data from smaller DASD to larger DASD, such as moving three (3) 3390-3 volumes onto a single 3390-9. Note that FDRMOVE cannot move data from 3380 to 3390 or vice versa. FDRMOVE only moves data sets that are on DASD and does not move archived or migrated data sets when the data sets are selected from the catalog.

Data sets can be moved between DASD volumes in the same DASD subsystem, or between DASD volumes in different subsystems and between different hardware manufacturers. FDRMOVE supports single- and multi-volume VSAM and non-VSAM data sets on SMS-managed or non SMS-managed volumes (SMS-managed data sets must be moved to SMS-managed volumes and non SMS-managed data sets must be moved to non SMS-managed volumes). FDRMOVE has an option to set the input volumes to DISNEW for SMS-managed volumes (disable new allocations) and PRIVATE for non SMS-managed volumes, so that new data sets go to the new output volumes; however, if you have a third-party DASD allocation product you may need to update its parameters.

FDRMOVE FUNCTIONS

FDRMOVE has three functions:

1. **FASTMOVE**

   Uses “instant” data set replication technology such as FlashCopy (on IBM and HDS subsystems) and EMCSNAP to quickly move data sets when the source and target data sets are in different DASD subsystems by moving the source volumes into the target subsystem. The user must reserve a number of offline “transit station” DASD volumes in the target subsystem. FDRMOVE uses FDRPAS to non-disruptively move source volumes to the transit station DASD volumes. Later, when the source data sets become inactive, FDRMOVE uses the instant replication to very quickly move the selected data sets to the new DASD volumes. When all data sets have been moved, the source volumes are automatically returned to their original DASD volumes.

   FASTMOVE can typically move up to 1TB of data per minute.

   FASTMOVE is particularly useful when moving data sets belonging to one or more applications, when those applications cannot be brought down for more than a few minutes. Once FDRPAS has moved the application volumes to the transit station DASD volumes, non-disruptively, you “bounce” the application, doing a very quick shutdown and restart (or de-allocating the application data sets briefly). FASTMOVE detects that the data sets are inactive (de-queued) and instantly move them while they are free.

   NOTE: Even if you do not have an instant replication hardware function on your target DASD subsystem, FASTMOVE can still be used to move data sets belonging to long-running applications. The data is copied with normal read/write I/O so the time that the application must be quiescent is longer, but large amounts of data can be copied quickly. By moving volumes into transit stations in the new control unit, the moves benefit from the enhanced performance of the control unit. Also you could leave the volumes in the transit stations and disconnect the old control unit.

2. **MOVE**

   Uses normal read/write I/O to move data sets when they become inactive (de-queued). MOVE immediately moves all data sets that are inactive. After that, it frequently checks the enqueue status of all selected data sets not yet moved. As soon as the data sets become inactive (such as a TSO user logging off), they are immediately moved. Because normal I/O is used, the time to move a data set is in proportion to its size.

   If the source and target DASD volumes for MOVE are in the same DASD subsystem, instant replication technology may be used if available.

3. **SIMULATION**

   FDRMOVE includes a simulation facility (SIMMOVE) that simulates a FASTMOVE step. SIMMOVE validates your control statements and the PASJOB internal job. SIMMOVE displays all the volumes that will be processed and shows how many volumes need to move to transit stations. SIMMOVE also displays all the selected data sets and shows which data sets are currently active (including the using job names).

MOVING TO A NEW DASD SUBSYSTEM

A primary function of FDRMOVE is consolidation of smaller DASD volumes onto larger DASD, such as 3390-9 to 3390-27, when the new DASD volumes are in a new DASD subsystem (control unit). FDRMOVE can also be used for load balancing, moving data within a subsystem, and other uses.

A combination of FASTMOVE and MOVE are used to accomplish this with minimal or no disruption to the use of the data sets. You can consolidate entire volumes, or subsets of the data sets on the volumes.

FASTMOVE requires that the new DASD subsystem be enabled for an instant replication technology, such as FlashCopy on IBM and HDS subsystems, or EMCSNAP (TimeFinder/Clone) on EMC subsystems. FASTMOVE also requires that the new subsystem have a number of unused offline DASD volumes to use as transit stations (temporary locations for volumes so that fast replication can be used).

The steps required to accomplish a MOVE to a new DASD subsystem:

1. The output volumes in the target subsystem must be prepared by initializing them with a volume serial number and a VTOC. You can do this using ICKDSF or an INNOVATION DATA PROCESSING utility provided with FDRMOVE called FDRINITV. The VTOCs on the larger target volumes probably need to be larger than those on the source volumes, since there will be more data sets on each volume.
2. If the target volumes are already initialized but have VTOCs that are too small, you can enlarge the VTOCs by using the EXPANDVTOC function of FDRPAS (see later in this section for more details).
3. If you plan to move some source volumes to target volumes using FDRPAS, you may again need to consider enlarging the VTOCs using EXPANDVTOC.
4. If you initialize new volsers, you simply add the new volumes to the same SMS storage group as the input volumes or update your storage management software appropriately.
5. Next, you execute the MOVE function of FDRMOVE to move data sets that are inactive or that become inactive while MOVE is running. You can leave MOVE running for hours or days, and it automatically recognizes selected data sets that have become free so that they can be moved with normal I/O.
6. The remaining data sets are probably those in use by long-running applications. FASTMOVE is used to move those data sets with instant replication very quickly, so that the time that the applications must be disrupted is very short. FASTMOVE submits a special FDRPAS job to move input volumes to and from the transit station DASD volumes in the new DASD subsystem. FASTMOVE then waits for data sets to become free (de-queued) just like MOVE above. When FASTMOVE is waiting on all needed data sets, you can “bounce” the application, taking it down and back up a short time after. FASTMOVE recognizes that the data sets are free and quickly move them to the target volumes by using the “instant replication” technology.
7. FDRMOVE terminates when:
   - All selected data sets have been moved
   - You enter a console STOP (P) command for the job
   - The time limit that is specified by the STOPAFTER= and/or STOPINACTIVE= operands is reached

NOTE: Even if you do not have an instant replication hardware function on your target DASD subsystem, the FASTMOVE or MOVE function of FDRMOVE can still be used to move data sets belonging to long-running applications. The data is copied with normal read/write I/O so the time that the application must be quiescent is longer, but large amounts of data are moved quickly.

325.1 FDRMOVE INTRODUCTION

SUMMARY

Both MOVE and FASTMOVE can be used to:

- Consolidate smaller volumes onto a larger volume, for SMS-managed and non SMS-managed volumes.
- Move additional smaller volumes to fill up the space on a larger volume with minimal or no disruption after using FDRPAS to move one volume to that larger volume non-disruptively. Note that the VTOC of the volume may need to be expanded.
- Move data sets to new volumes or SMS storage groups with minimal or no disruption.
- Fewer UCBs are required to address the data, so freed UCBs can be used for other purposes.
- Implement load balancing - moving data sets or volumes to improve performance.

MOVE is used for data sets that are naturally de-allocated for a period of time, so the data is moved with normal read/write I/O.

FASTMOVE is used for data sets that can only tolerate a brief disruption, so data sets are copied with fast replication functions such as FlashCopy and EMCSNAP. The data rate depends on the configuration, the number of FASTMOVE subtasks, the number of data sets to be moved, and other factors. FASTMOVE can typically move 250 data sets, occupying up to 1TB of data, that reside on 16 volumes in about 1 minute. FASTMOVE can typically allocate, catalog, and move over 1000 small data sets per minute with 16 concurrent volumes. Moving 16 concurrent volumes requires two FDRMOVE jobs running concurrently.

PRODUCT DEMOS

Product demos are available on the INNOVATION DATA PROCESSING web site for FDRMOVE. Go to web site http://fdr.com and click the FDRMOVE link under the "DEMOS" icon.

The direct link for the FDRMOVE Product Demo is: http://www.fdr.com/demo.cfm

USING FDREPORT WITH FDRMOVE

We have created a set of FDREPORT jobs to identify potential problems before you attempt DASD consolidation (for example, 3390-9 to 3390-54 or to EAV) with FDRMOVE. FDREPORT can assist you in resolving issues before you start the migration and allows you to easily meet your SLA. Contact INNOVATION DATA PROCESSING for a temporary license of FDREPORT if you are not currently licensed for FDREPORT.

The sample **FDREPORT_FDRMOVE_EXAMPLES** can be found in the download directory under **Tech_Support_Samples**.

Some of these reports:

- Check the available space in the VTOC, VTOCIX, and VVDS.
- Identify volumes with catalogs, sysres volumes, PAGE and SWAP data sets.
- Report data sets cataloged to the wrong volume or not cataloged.
- Show missing or orphan VVR/NVRs and identify multi-volume data sets.
- List data sets in an SMS storage group and high-level qualifier.
- List volumes that have linklst data sets.

325.2 FDRMOVE JCL AND STATEMENTS

FDRMOVE JCL

The JCL to execute FDRMOVE is very simple. Here is a sample:

```
//MOVE     EXEC PGM=FDRMOVE,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//FDRSUMM  DD SYSOUT=*
//SYSIN    DD *
  Control statements
/*
//PASJOB   DD DATA,DLM=$$
  FDRPAS JCL and control statements for FASTMOVE
$$
```

RECOMMENDATION: *REGION=0M is always recommended; in order to process multiple input volumes concurrently, a large below-the-line region is needed.*

The STEPLIB DD statement points to the authorized library containing FDRPAS and FDRMOVE.

WARNING: ***For EMC SNAP, FDR invokes an EMC API to request the SNAPs. The current EMC TimeFinder software must be in the system linklist or you can concatenate the proper EMC library to the FDRMOVE STEPLIB.***

The FDRSUMM DD statement contains a summary of volumes processed, one line per volume. Note that some volumes appear more than once if active data sets cause the volume to be processed repeatedly.

The PASJOB DD statement points to a model job stream to execute the special FDRPAS transit station job that is required for FASTMOVE. This DD statement is optional but is highly recommended. It is used only for a FASTMOVE operation; it is ignored for a MOVE operation. Details of the job stream are discussed later in this section. The DD statement can point to a sequential data set on DASD or a member of a PDS or PDSE. If it is an in-stream input, you must code the DD statement exactly as shown above and follow the job stream with “$$” in columns 1 and 2 to indicate the end of the job.

FDRMOVE Control Statements

The control statements for FDRMOVE are also simple. The operand descriptions below include those operands that are unique to FDRMOVE or essential to the use of FDRMOVE. In addition, FDRMOVE accepts many operands that are documented for FDRCOPY in Chapter 21 of the regular FDR product manual, but these are rarely needed for FDRMOVE.

FDRMOVE is intended to move data sets without renaming them, so do not specify the NEWNAME=, NEWG=, or NEWI= operands on a SELECT statement.

MOVE, FASTMOVE, and SIMMOVE Statement Syntax

The syntax of the MOVE, FASTMOVE, and SIMMOVE statements are:

```
MOVE                TYPE=DSF
FASTMOVE
SIMMOVE
                    ,DATA=ALL|                  ,SCANDELAY=nnn
                          USED
                    ,DISABLENEW=YES|            ,SELTERR=YES|
                                NO                        NO
                    ,EATTR=OPT                  ,STGADMIN
                    ,MAXCARDS=1024|             ,STOPINACTIVE=nnnn
                              nnnnn
                    ,MAXTASKS=8|                ,STOPAFTER=nnnn
                              n
                    ,REMOTEINSTANT=PREFERRED|   ,VOLSORT=YES|
                                   REQUIRED                 NO
                                                ,VTOCEMPTY=CHECK|
                                                           NOCHECK
```

MOVE, FASTMOVE, and SIMMOVE Statements

The first statement input to FDRMOVE must always be a MOVE, FASTMOVE, or SIMMOVE statement.

MOVE – Invokes the MOVE function. See Section 325.3 "FDRMOVE MOVE Function with Examples".

FASTMOVE – Invokes the FASTMOVE function. See Section 325.4 "FDRMOVE FASTMOVE Function with Examples".

SIMMOVE – Performs a simulation of a FASTMOVE step. Control statements are validated and the data sets you specified are selected and displayed. The internal "FDRPAS Transit JOB for FASTMOVE" in Section 325.2 is submitted to be sure that it works (no JCL or control statement errors); as soon as it starts successfully it is stopped. No data is moved by the simulation.

The operands for MOVE, FASTMOVE, and SIMMOVE are:

MOVE, FASTMOVE, and SIMMOVE Statement Operands

TYPE=DSF

Specifies that a data set move is to be performed. This operand is required.

DATA=

ALL – All allocated tracks of all data sets are moved.

USED – Only the used portion of Physical Sequential (PS) and Partitioned Organization (PO, PDS) data sets are moved. Even though only used tracks are copied, the output data sets retain the size (allocation) of the input data sets. On volumes with many PS and PO data sets, DATA=USED makes the move run faster.

Default: USED.

DISABLENEW=

NO – A quiesce is not done on the source volumes and new allocations can continue to go to those volumes as well as the output volumes.

YES – SMS-managed source volumes are marked as DISNEW (disabled for new allocations). Non SMS-managed volumes are changed to mount status PRIVATE if they are currently PUBLIC or STORAGE. Changing the mount status to PRIVATE prevents new allocations to those volumes unless the specific volume serial number is provided in the user's JCL.

Default: NO.

The output volumes should be defined to the same SMS storage group (or esoteric unit name for non SMS-managed volumes) as the input volumes, so new allocations naturally go to the new volsers. `DISABLENEW=YES` is recommended if you want to discontinue use of the source volumes, such as when you are migrating all data off of a DASD subsystem.

NOTE: FDRMOVE does not re-enable the source volumes if you use DISABLENEW=YES, assuming that you want to stop using them. You must manually re-enable the volumes for new allocations if you want to continue using them.

WARNING: ***If you activate a new SMS configuration, you must be sure that the volumes are marked as DISNEW in the SCDS before you activate it. Otherwise, the volumes may return to ENABLE status, allowing new allocations to the volumes. Also, if you have separate SMS configurations for each LPAR, `DISABLENEW=YES` only disables the volumes on the current LPAR, so you are responsible for manually disabling the volumes on the other LPARs before moving any data sets.***

EATTR=OPT

Allows FDRMOVE to allocate non-VSAM data sets in cylinder-managed space on Extended Address Volumes (EAVs), unless the DSCB for the data set specifies `EATTR=NO`. (VSAM data sets are eligible by default to be allocated in cylinder-managed space, and are not affected by this operand.) See "Extended Address Volumes (EAVs)" in Section 325.11 for additional information on how data sets are allocated on Extended Address Volumes (EAVs).

MAXCARDS=

***nnnnn* –** Accept additional `SELECT` and `EXCLUDE` statements (over 1024).

Default: 1024 statements.

WARNING: ***If you use `SELECT CATDSN=, VOL=` ending with an asterisk, or `STORGRP=`, FDRMOVE generates internal `SELECT` statements, one for each source DASD volume selected. If more than 1024 volumes are selected, `MAXCARDS=` must be specified to increase the control statement table size. Very large values for `MAXCARDS=` may cause storage shortages; contact INNOVATION DATA PROCESSING if you need assistance.***

MAXTASKS=

n **–** The maximum number of input DASD volumes that are processed concurrently. SYSPRIN*x* DD statements are dynamically allocated to SYSOUT to receive messages from the FDRMOVE subtasks. Selected subtask messages are also repeated on SYSPRINT when each subtask ends.

Default: 8.

WARNING: ***FDRMOVE automatically reduces the default value to 7 if the below-the-line private area is 9 MB, or to 6 if the below-the-line private area is 8 MB or smaller. Also, FDRMOVE reduces MAXTASKS dynamically if S878 ABENDs occur during execution.***

REMOTEINSTANT=

PREFERRED – An attempt is made to utilize the remote pair FlashCopy feature. If the attempt is not successful, the MOVE operation continues and REMOTEINSTANT is ignored. If FlashCopy is available for the MOVE operation, it is used and the metro mirror relationship may enter a "pending" state.

REQUIRED – An attempt is made to utilize the remote pair FlashCopy feature. If the attempt is not successful, the MOVE operation continues, but does not use FlashCopy.

NOTE: If the volumes are not in a metro mirror relationship setup for remote mirror FlashCopy, the REMOTEINSTANT operand is ignored.

SCANDELAY=

nnn **–** The number of seconds (1 to 255) between scans of the list of active data sets. When inactive data sets are found, the volumes they reside on are processed to move the data sets. When this is complete or if no inactive data sets were found, FDRMOVE waits for `SCANDELAY=` seconds before testing again.

Default: 15 seconds for MOVE, 2 seconds for FASTMOVE.

SELTERR=

Specifies what happens at step termination if FDRMOVE finds that a SELECT statement was never referenced (no data set on any input DASD volume was selected by the statement).

NO – A non-zero return code is **not** to be set at step termination. You might use `SELTERR=NO` when you expect some unmatched SELECT statements, perhaps because some data sets may not exist.

YES – A non-zero return code is set at step termination to call attention to a possible control statement error.

Default: YES.

STGADMIN

If specified, FDRMOVE checks to see if the user ID under which it is running is authorized to security class FACILITY profile name STGADMIN.ADR.STGADMIN.MOVE. If so, all security checks are bypassed for all FDRMOVE operations, including system allocation and catalog functions. Using STGADMIN allows FDRMOVE to move all selected data sets without needing any additional security.

Default: FDRMOVE does normal security checks including system allocation and catalog security. The user ID under which FDRMOVE is running **MUST** be authorized to create and update **ALL** selected data sets. The security system overhead may be significant if many data sets are involved.

RECOMMENDATION: ***INNOVATION DATA PROCESSING strongly recommends the use of the STGADMIN.ADR.STGADMIN.MOVE profile for all FDRMOVE operations, especially if you have separate security configurations. STGADMIN reduces the CPU time used by FDRMOVE.***

STOPINACTIVE=

nnnn **–** The number of idle-time minutes that FDRMOVE accumulates before terminating automatically. Idle time is not incremented when FDRMOVE is actively moving data sets, only when it is waiting for data sets to become unallocated.

Default: FDRMOVE executes indefinitely, until all selected data sets have been moved or a console STOP (P) or CANCEL (C) is issued for the FDRMOVE job.

STOPAFTER=

nnnn **–** A number of elapsed time minutes after which FDRMOVE terminates automatically, after finishing all currently active moves (active FDRMOVE subtasks).

Default: FDRMOVE executes indefinitely, until all selected data sets have been moved or a console STOP (P) or CANCEL (C) is issued for the FDRMOVE job.

STOPAFTER= can be used when you want FDRMOVE to execute only during certain times. For example, STOPAFTER=240 terminates FDRMOVE approximately four hours after the job is started.

VOLSORT=

NO – Output volumes are not rotated. The output volumes are used in the order the volumes are specified by NVOL= or selected from the storage group. You can use VOLSORT=NO when you want to control where data sets are placed.

YES – Output volumes (NVOL= or NEWSTORGRP= on the SELECT statements) are rotated as described under "Output Volume Selection for FDRMOVE" in Section 325.2. When data sets are moved from multiple input volumes, FDRMOVE spreads the data sets by moving different input volumes to a different output volume (NVOL=).

Default: YES. INNOVATION DATA PROCESSING recommends that you do not change the default, especially if STORGRP= or NEWSTORGRP= is specified on a SELECT statement.

WARNING: ***When you specify VOLSORT=NO many checks on the NVOL= list are bypassed; it is your responsibility to specify proper online target volsers.***

VTOCEMPTY=

Is only accepted on a SIMMOVE statement.

CHECK – If used with `SELECT ALLDSN,VOL=`, It checks the VTOCs of all volumes specified on `MOUNT` statements, and show you the volumes that still have data sets on them (other than the VTOC, VTOCIX, and VVDS). `VTOCEMPTY=CHECK` can be used to check the input volumes after a MOVE or FASTMOVE to see if they have been emptied. Neither `NVOL=` nor `NEWSTORGRP=` is required.

If used with `SELECT CATDSN=`, you must provide `NVOL=` or `NEWSTORGRP=` for all potential target volumes for the selected data sets and it displays all those data sets that are not on volumes in the `NVOL=` list. In other words, it displays the data sets that still need to be moved.

NOCHECK – VTOCs are not checked.

Default: NOCHECK

EXCLUDE STATEMENT

One or more EXCLUDE statements can optionally be input to exclude certain data sets from selection. EXCLUDE statements should precede SELECT statements. EXCLUDE statements are rarely needed unless you want to exclude by data set size (SIZE=). Only a few operands apply to EXCLUDE in FDRMOVE. Some EXCLUDE statements are automatically generated by FDRMOVE. See Section 325.7 "FDRMOVE Consolidate Volumes with MOVE Function".

EXCLUDE STATEMENT SYNTAX

The syntax is:

```
EXCLUDE|     ALLDSN
X            DSN=filter
             ,DSORG=(DA                  ,SIZE=nnnnn
                    ,EF
                    ,PO                  ,VOL=vvvvvv
                    ,PS
                    ,UM
                    ,UN)
```

EXCLUDE STATEMENT OPERANDS

The operands are:

ALLDSN

Excludes all data sets, so it must be used with one or more of DSORG, SIZE, or VOL to limit the data sets to be excluded.

DSN=

filter **–** Specifies a fully-qualified data set name or a filter to be used for generic data set exclusion. See the explanation under SELECT below.

DSORG=

Specifies that the EXCLUDE statement is to apply only to data sets whose data set organization matches one of the DSORG values specified. If more than one DSORG is specified, they must be enclosed in parentheses.

Valid DSORG values are:

DA – BDAM

EF – VSAM (including DB2 and Linear)

PO – Partitioned

PS – Sequential

UM – Unmovable

UN – Undefined (NONE)

SIZE=

nnnnn **–** Specifies a data set size threshold, in tracks. Data sets above this size are excluded.

INNOVATION DATA PROCESSING has found that the elapsed time to invoke instant replication such as FlashCopy and EMCSNAP is often higher than the time to do normal read/write I/O for small data sets. EXCLUDE ALLDSN,SIZE=1500 might be used in a MOVE step to move data sets under 100 cylinders with normal I/O, while larger data sets can be moved with instant replication in a FASTMOVE step. Examples are shown later.

The maximum value that can be specified for SIZE= is 65535.

VOL=

vvvvvv **–** Specifies the input DASD volume serial(s) to which the EXCLUDE statement is to apply. VOL= may specify a single serial (for example, VOL=ABC123) or a group of volumes all starting with the same prefix (for example, VOL=ABC*). If VOL= is omitted, it applies to all input volumes.

SELECT STATEMENT

One or more SELECT statements must be input to FDRMOVE. SELECT specifies the data sets and/or volumes to be moved, and additional operands can limit the data sets.

SELECT STATEMENT SYNTAX

The syntax is:

```
SELECT      ALLDSN|
S           CATDSN=filter|
            DSN=filter
            ,DATA=ALL                   ,NVOL=(vol[,vol[,…]])
                                            [,NEWCUID=sssss]|
            ,DSORG=(DA                  ,NEWSTORGRP=storgrp
                   ,EF                      [,NEWCUID=sssss]|
                   ,PO                  ,ENEWSTORGRP=storgrp
                   ,PS
                   ,UM                  ,STORCLAS=storclas
                   ,UN)
                                        ,STORGRP=storgrp
            ,MGMTCLAS=mgmtclas              [,CUID=sssss]

            ,NEWSGFREE%=nn              ,VOL=vvvvvv
```

SELECT STATEMENT OPERANDS

The operands are:

ALLDSN

Selects all data sets on a volume, so it must be used with VOL=. The VTOCIX, VVDS, and ABR Model DSCB are automatically excluded.

CATDSN=

filter **–** Specifies a fully-qualified data set name or a filter to be used for generic data set selection from system catalogs. The filter syntax is the same as for DSN= below, but system catalogs are searched for the data sets. The volume serial of each data set is taken from the catalog, so `VOL=` is not required. However, `VOL=` can be specified to limit the cataloged data sets to only the volume(s) given.

FDRMOVE uses the information from the catalog to build internal `SELECT DSN=` statements, one for each input volume found.

DSN=

filter **–** Specifies a fully-qualified data set name or a filter to be used for generic data set selection from the VTOCs of the input volume(s). It must be used with `VOL=` to specify the input volume(s).

The filter is extremely flexible and is completely described in Section 80.14 of the FDR product manual. Here are the filter characters most likely to be used with FDRMOVE:

/ (*slash*) – Represents exactly one character, with any value.

***** (*asterisk*) – Represents a variable number of characters in an index level of a data set name (from zero to eight characters).

****** (*double asterisk*) – Represents a variable number of characters that may span more than one index level.

Examples:

```
DSN=ABC**          selects all data sets beginning with ABC.
DSN=ABC.**         selects all data sets with a 1st index
                   level of ABC.
DSN=ABC.Z*.CNTL    selects all data sets with a 1st index of
                   ABC, a 3rd index of CNTL and a 2nd index
                   beginning with Z.
DSN=ABC///.**      selects all data sets with a 6-character
                   1st index starting with ABC.
```

DATA=

ALL – All allocated tracks of all data sets that are selected by this SELECT statement are copied.

Default: The value of DATA= on the MOVE or FASTMOVE statement (that defaults to moving only used tracks for PS and PO data sets).

DSORG=

Specifies that this SELECT statement is to apply only to data sets whose data set organization matches one of the DSORG= values specified. If more than one DSORG= is specified, they must be enclosed in parentheses.

Valid DSORG= values are:

DA – BDAM

EF – VSAM (including DB2 and Linear)

PS – Sequential

PO – Partitioned

UM – Unmovable

UN – Undefined (NONE)

MGMTCLAS=

***mgmtclas* –** Specifies the SMS management class to be associated with the output data set, overriding the original management class (if any).

Default: The SMS management class of the input data set (if any) is used for the output.

NOTE: The management class Automatic Class Selection (ACS) routine is not called. The specified management class name is not checked for validity.

NEWSGFREE%=

***nn* –** Specifies the percentage of free space to leave on the target volume when moving with the NEWSTORGRP= or ENEWSTORGRP= operands.

Default: The value is dynamic based on the entire storage group so that data is spread across all volumes in the storage group evenly.

NVOL=

***vol* –** Specifies the volume serial(s) of output DASD volumes where data sets selected by this statement are to be moved. It is required. You may specify:

- A single specific volume serial, for example,

  `NVOL=ABC123`

- A list of specific volume serials, enclosed in parentheses, for example,

  `NVOL=(TSO001,TSO002,TSO003)`

- A group of volumes by placing an asterisk at the end of the volume serial number prefix, for example,

  `NVOL=TSO*`

- A combination of specific and group, for example,

  `NVOL=(TSO*,PROD*,ABC001)`

A group is resolved by scanning all online DASD volumes for the specified volume serial number prefix, but the order of the volumes is unpredictable. Only the first 255 volumes selected are used. Internally, specific volsers are moved to the front of the list, followed by any prefixes.

The output volumes are dynamically allocated when the volumes are selected as a target.

The `NVOL=` list may contain a mix of SMS-managed and non SMS-managed volumes. For each input volume, only those NVOLs matching the type of the input volume (SMS-managed or non SMS-managed) are chosen. If the `NVOL=` list includes both 3380s and 3390s, only those matching the type of the input volume are selected. Only online volumes are selected; if no specified volumes are online, you receive a diagnostic message.

`NEWSTORGRP=`, `ENEWSTORGRP=`, and `NVOL=` are mutually exclusive and only one of these operands can be specified on a SELECT statement.

For an explanation of how `NVOL=` is used, see "Output Volume Selection for FDRMOVE" in Section 325.2.

NEWSTORGRP=

***storgrp* –** Specifies an SMS storage group name for output data set allocation. All online volumes in the storage group become an implied `NVOL=` operand. The volumes are selected and sorted in descending order by free space reported by SMS. `NEWSTORGRP=`, `ENEWSTORGRP=`, and `NVOL=` are mutually exclusive and only one of these operands can be specified on a SELECT statement.

NEWSTORGRP= is used like `NVOL=` for output volume selection.

NEWCUID=*sssss* – (five digits) is an additional operand that can be specified to limit the output `NEWSTORGRP=` selection to a specified DASD control unit. `NEWCUID=` specifies the 5-digit serial number of the control unit. The `NEWCUID=` operand must precede the `NEWSTORGRP=` operand, for example, `NEWCUID=12345,NEWSTORGRP=ABC`. You must use `NEWCUID=` when you are moving data sets to the same storage group (the storage group has already been updated to include the target DASD volumes), so that only volumes in the target control unit are selected. You may need to use `NEWCUID=` when the new storage group contains volumes from multiple control units, to limit the selection to one control unit.

You must specify `NEWCUID=` if the source volumes are in the same storage group as the target volume.

NOTE: FDRMOVE allocates to volumes in the new storage group even if they are marked as DISNEW or QUINEW. Keeping volumes in DISNEW status allows FDRMOVE to allocate the data sets to the volumes in this status, and prevents users from using them. Use `ENEWSTORGRP=` to bypass volumes marked as DISNEW from the allocate volume list.

ENEWSTORGRP=

***storgrp* –** Specifies an SMS storage group name for output data set allocation in the same manner as NEWSTORGRP=, but does not select volumes that are marked as DISNEW. All online volumes in the storage group become an implied NVOL= operand. The volumes are selected and, sorted in descending order by free space reported by SMS. NEWSTORGRP=, ENEWSTORGRP=, and NVOL= are mutually exclusive and only one of these operands can be specified on a SELECT statement.

ENEWSTORGRP= is used like NVOL= for output volume selection.

NEWCUID=*sssss* – (five digits) is an additional operand that can be specified to limit the output ENEWSTORGRP= selection to a specified DASD control unit. NEWCUID= specifies the 5-digit serial number of the control unit. The NEWCUID= operand must precede the ENEWSTORGRP= operand, for example, NEWCUID=12345,ENEWSTORGRP=ABC. You must use NEWCUID= when you are moving data sets to the same storage group (the storage group has already been updated to include the target DASD volumes), so that only volumes in the target control unit are selected. You may need to use NEWCUID= when the new storage group contains volumes from multiple control units, to limit the selection to one control unit.

You must specify NEWCUID= if the source volumes are in the same storage group as the target volume.

STORCLAS=

***storclas* –** Specifies the SMS storage class to be associated with the output data set, overriding the original storage class.

Default: The SMS storage class of the input data set is used for the output.

NOTE: The storage class Automatic Class Selection (ACS) routine is not called. The specified storage class name is not checked for validity.

STORGRP=

***storgrp* –** Specifies an SMS storage group name for input volume selection. All volumes in the storage group become implied VOL= operands. FDRMOVE internally generates multiple SELECT statements, one for each volume, so MAXCARDS= may need to be increased if you select many volumes. STORGRP= can only be used with SELECT ALLDSN or DSN=, not CATDSN=. You must not specify both STORGRP= and VOL= on the same SELECT statement. The volumes are sorted in descending order by allocated space to be matched with the sorted target volumes specified by NEWSTORGRP= or ENEWSTORGRP=.

CUID=*sssss* – (five digits) is an additional operand that can be specified to limit the input STORGRP= selection to a specified DASD control unit. CUID= specifies the 5-digit serial number of the control unit. The CUID= operand must precede the STORGRP= operand, for example, CUID=12345,STORGRP=ABC. You must use CUID= when the storage group has already been updated to include the target DASD volumes, so that only volumes in the source control unit are selected. You may need to use CUID= when the storage group contains volumes from multiple control units, to limit the selection to one control unit.

NOTE: The CUID= operand cannot be specified unless the STORGRP= operand is also specified.

You must specify CUID= if the source volumes are in the same storage group as the target volume.

You can display the serial number of a control unit with the console command "DS QD,01C0" specifying a device in the control unit.

DS QD,01C0

```
IEE459I DEVSERV QDASD
UNIT VOLSER SCUTYPE DEVTYPE   CYL  SSID SCU-SERIAL DEV-SERIAL
01C0 SYM000 2105F20 2105    10017  3000 XX07-00547 XX07-00547
```

To exclude one or more volumes from the STORGRP= selection, you can specify an EXCLUDE ALLDSN,VOL=*volser* statement before the SELECT.

VOL=

***vvvvvv* –** Specifies the input DASD volume serial(s) to which this SELECT statement is to apply.

VOL= can be a single volume serial number (for example, VOL=ABC123) or a group of volsers, terminated with an asterisk (VOL=ABC*). VOL=* is not valid.

For CATDSN=, VOL= limits the volumes that are selected from the catalog; it is optional.

Care should be take when specifying VOL=ABC* since it may select many unintended volumes. Run with SIMMOVE to verify the volumes to be selected.

You must not specify both STORGRP= and VOL= on the same SELECT statement.

OUTPUT VOLUME SELECTION FOR FDRMOVE

FDRMOVE does not use SMS (or other allocation control software) to determine data set placement, so you must specify NVOL=, NEWSTORGRP=, or ENEWSTORGRP= even if you are moving SMS-managed data sets. You may need to update allocation control software, other than SMS, to prevent allocation to the input volumes and to allow allocation to the new volumes.

- For non SMS-managed input volumes, FDRMOVE selects only non SMS-managed target volumes from the NVOL= list you provide.
- For SMS input volumes, FDRMOVE selects only SMS target volumes from the NVOL= list you provide. However, if all the input volumes are SMS, it is highly recommended that you use the NEWSTORGRP= or ENEWSTORGRP= operand to select the target volumes.

If the NVOL list contains both SMS-managed and non SMS-managed volumes or 3380s and 3390s, FDRMOVE selects only those output volumes that match the type of each input volume. The target volumes may be the same size as the input, or can be larger or smaller.

If you have several consecutive SELECT statements (specifying DSN= or ALLDSN) **with the same NVOL= list**, FDRMOVE round-robins (**"**rotate**"**) the list. On the second SELECT the first volume is moved to the last slot, and this is repeated for subsequent SELECT statements. This round-robin rotation spreads the data sets across the output volumes. Since each SELECT specifies one input volume, the data sets from one input volume are moved to the first NVOL=, data sets from the second input go to the second NVOL=, and so on. If the NVOL= lists are different on two consecutive SELECT statements, the rotation is restarted (by using the new NVOL=).

SELECT CATDSN= – Creates internal SELECT DSN= statements, one for each unique data set and per input volume found in the catalog, each with the same NVOL= list copied from the SELECT CATDSN=. So the output volume rotation is always done for each CATDSN= statement selecting multiple input volumes.

SELECT VOL=ABC* – (A volume serial group ended with an asterisk) creates internal SELECT DSN= statements for each input volume found, each with the same NVOL= list that is copied from the SELECT VOL=ABC*. So the output volume rotation is always done for each VOL=ABC* statement selecting multiple input volumes.

SELECT STORGRP= – Creates internal SELECT VOL= statements, one for each input volume found in the storage group, each with the same NVOL list copied from the SELECT. So the output volume rotation is always done for each STORGRP= statement selecting multiple input volumes.

The rotation described above is not done if VOLSORT=NO is specified on the main statement. VOLSORT=NO should not be specified with CATDSN= or STORGRP=. The purpose of VOLSORT=NO is to allow you to direct a specific SELECT VOL= statement to a specific NVOL= list.

SELECT NEWSTORGRP= or ENEWSTORGRP= initially sorts the volumes in the storage group by available free space, these target volumes are then matched with source volumes that are initially sorted by available free space.

Pieces of multi-volume data sets always go to the first volume in the rotated NVOL list, unless a piece of the data set already exists there, in that case other NVOLs in the same list are tried.

If CATDSN= selects only one input volume or an ALLDSN/DSN= is specified with a unique NVOL= list, the first volume in the NVOL= list (or the first found by the UCB scan if a volume serial number prefix was specified) is the primary output volume. All data sets go to that primary volume unless insufficient free space is available or the data set is multi-volume (see below). If the allocation fails, other volumes in the NVOL= list are tried. If you are moving a multi-volume data set (non-VSAM or SMS-managed VSAM), the volume sequence number of the piece of the data set being allocated is used to select a specification from your list. For example, if NVOL=(A,B,C), the second piece of the data set goes to volume "B". If that specification is a group, the first volume in the UCB chain matching that group is tried.

The NVOL= list for FDRMOVE should specify more than one volume serial number for two reasons:

1. Multi-volume data sets must be moved to the same number of volumes that they currently occupy. FDRMOVE can move each piece of a multi-volume data set independently but it cannot combine pieces; so if the selected data sets includes, for example, a three volume multi-volume data set you must have at least three volumes in the NVOL= list.
2. Although you may be attempting to merge several smaller DASD volumes into one larger DASD volume, it is possible that because of volume fragmentation or the order of allocation, all data sets from the input volumes may not fit onto one output volume. The NVOL= list should contain extra volumes as spill volumes.

A SIMMOVE displays, for each input volume, the highest multi-volume sequence number found on that volume (all non-VSAM files and SMS VSAM only). Find the highest displayed value for any volume for data sets going to the same NVOL= list, and you know the minimum number of volumes that must be in the list.

NOTE: If an allocation is attempted on several volumes from your NVOL= list but it fails on all of them, the message printed usually shows the allocation failure codes from the first volume only; failure codes from other volumes are not displayed and may be different.

CHOOSING THE NUMBER OF TARGET VOLUMES

Here are some guidelines for specifying the NVOL= list.

- You wish to consolidate 60 full 3390-3s to empty 3390-9s. Because the capacity of the 3390-9 is three times that of the 3390-3, you need a minimum of (20) 3390-9 volumes in the NVOL= list. However, you usually want to specify a few extra NVOL= volumes. INNOVATION DATA PROCESSING recommends 10% extra, which would be two volumes in this example, for a total of 22. The first 22 input volumes go to the 22 output volumes, the next 22 go to the same 22 output volumes, and the last 16 would go to the first 16 output volumes. If the allocation of any data set fails on its target volume, FDRMOVE moves down the NVOL= list until it finds a successful volume for the data set. The extra output volumes help ensure that the allocations are successful.
- The same formula applies regardless of the size of the input and output DASD volumes. Use the ratio of the DASD sizes to determine the number of output volumes. For example, a 3390-3 to a 3390-27 would be a 9:1 ratio.
- If you determine that the average allocation on your input volumes is less than 80%, you may be able to reduce the number of NVOLs.
- If you first used FDRPAS to move volumes from the larger DASD to the smaller DASD, only two thirds of the smaller DASD would then remain to be moved by FDRMOVE, and the resulting ratio for 3390-3 to 3390-9 would be 2:1, and for a 3390-3 to 3390-27 would be 8:1.
- If the input DASD volumes contain multi-volume data sets, the NVOL= list must contain at least as many volumes as the maximum number of volumes occupied by any one data set. A SIMMOVE displays, for each input volume, the highest multi-volume sequence number found on that volume (non-VSAM and SMS VSAM only). Find the highest displayed volume for data set going to the same NVOL= list, and you know the minimum number of volumes that must be in the list.
- If you select data sets with CATDSN=, use the SIMMOVE function to determine the number of input volumes.

FDRPAS TRANSIT JOB FOR FASTMOVE

FASTMOVE requires a special FDRPAS "TYPE=TRANSIT" job to be active on the same system as the FDRMOVE job. This job handles the task of moving input volumes to and from transit stations. Although you can submit this job manually, INNOVATION DATA PROCESSING highly recommends that you let FASTMOVE automatically submit the FDRPAS job when it is needed. FASTMOVE also automatically stops the FDRPAS job when it is done.

If FASTMOVE determines that some of the input volumes are not already in the target subsystem, then FASTMOVE submits the FDRPAS job with the identified volume serials. The FDRPAS job starts moving input volumes to the transit stations. It processes DASD volumes concurrently, up to the maximum number of tasks (MAXTASKS=). As each volume completes, another volume is chosen to move until all are moved or the number of transit station devices is exhausted. When FASTMOVE finishes moving all selected data sets from an input volume, FDRPAS automatically moves it back to its original device. Another volume can then be moved to that transit station.

In order for FASTMOVE to submit the FDRPAS job, you must include the optional PASJOB DD statement in the FASTMOVE step. This DD statement must point to a model FDRPAS transit station (SWAP TYPE=TRANSIT) job stream, including JOB statement, other JCL and control statements.

If FASTMOVE submits the FDRPAS job, it monitors it to be sure that it becomes active. If not, the FASTMOVE step is terminated. At first, FASTMOVE waits up to five minutes for the FDRPAS job to start, with a status message every 15 second warning that it has not started. If the FDRPAS job never starts, the FASTMOVE step is terminated. You should ensure that the FDRPAS job has a high priority so it starts promptly. A SIMMOVE only waits 15 seconds for the FDRPAS job to start.

The FDRPAS job stream must be a complete job, including a valid JOB statement. This is a "job within a job", but the FDRPAS job stream is submitted (via the internal reader) for execution only if FASTMOVE needs to move a volume to a transit station. Although FDRMOVE does some validation on the JOB statement, it is not comprehensive. If the FDRPAS job does not start, check the JOB statement (SYSLOG may contain some JES messages that may give a clue as to the error). Here is a sample FDRPAS job JCL:

```
//TRANSIT1 JOB (accounting info),FDRMOVE,CLASS=c,
//         MSGCLASS=m  <-- add any other required JOB parameters
//TRANSIT  EXEC PGM=FDRPAS,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//FDRSUMM  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  Control statements
/*
```

If you execute multiple concurrent FDRMOVE steps, then you need a unique job name for every transit job so that they can also run concurrently. The job must execute on the same LPAR as the FASTMOVE job.

RECOMMENDATION: *REGION=0M is always recommended; in order to process multiple input volumes concurrently, a large below-the-line region is needed.*

The STEPLIB points to the authorized library containing FDRPAS and FDRMOVE.

TRANSIT JOB SWAP STATEMENT

The first control statement in SYSIN for the FDRPAS job (after the JCL) must be a `SWAP TYPE=TRANSIT` statement.

TRANSIT JOB SWAP STATEMENT SYNTAX

The syntax of the SWAP statement is:

```
SWAP    TYPE=TRANSIT
        ,#SYSTEMS=nnn                   ,PACEDELAY=0|
                                                   nnnnn
        ,MAXACTIVESWAPS=NO|             ,PACING=DYNAMIC|
                        YES                        STATIC
        ,MAXCARDS=250|                  ,TRANSITRETURN=NO|
                  nnnn                                 YES
        ,MAXTASKS=5|
                  nn
```

Most of these operands are the same as the operands documented for the FDRPAS SWAP statement (see Section 310.2 "SWAP, SWAPDUMP, SIMSWAP, SIMSWAPMON Statements"). Please see that section for the explanation of #SYSTEMS=, PACEDELAY=, and PACING=. Also note that `NONRESPONDING=RETRY` is forced, so any non-responding FDRPAS MONITOR tasks cause message FDRW68 to be issued, where you can retry the failing systems or cause the transit station swap to fail. Also, `LARGERSIZE=OK` is set so that transit station devices can be larger than the source volumes.

TRANSIT JOB SWAP STATEMENT OPERANDS

The other operands are:

TYPE=TRANSIT

Is required.

MAXACTIVESWAPS=

NO – The total number of active transit SWAP tasks, in multiple FDRPAS jobs, is not limited or controlled. If you submit multiple FDRPAS jobs, each of them may be actively copying data for multiple DASD volumes (up to the `MAXTASKS=nn` limit in each job). Many such jobs may overload DASD channels and system common storage. In this case, INNOVATION DATA PROCESSING recommends that you submit a few such jobs at one time, and submit new jobs as each old job finishes.

YES – FDRPAS limits the number of transit SWAP tasks in multiple FDRPAS jobs that can be in the initial "pass 1" copy phase, where the tracks that are identified by FDRPAS as "in-use" are copied. The limit is equal to the value of `MAXTASKS=nn` (see below) which has a maximum of 32. If you submit multiple FDRPAS jobs, FDRPAS limits to "*nn*" the number of volumes that can be in "pass 1" across all of those jobs. This allows you to submit many FDRPAS jobs while limiting the number of volumes that are actively copying data. Beyond "pass 1", FDRPAS periodically copies updated tracks that are a much smaller load on the system.

Default: NO.

MAXCARDS=

***nnnn* –** Specifies the maximum number of MOUNT statements that can be present in this FDRPAS step, from 1 to 9999. There is one MOUNT for each input volume that needs to be moved to transit stations, so set the value appropriately. Note that FDRPAS acquires a table with a size of 160*MAXCARDS bytes in below-the-line storage, so very large values may cause GETMAIN failures.

Default: 250.

MAXTASKS=

***nn* –** Specifies the maximum number of volumes that can be moving to or from transit stations concurrently by this SWAP task, from 1 to 64. You must follow the SWAP statement with multiple MOUNT statements, specifying the volumes to be processed (the MOUNT statements may be generated by FASTMOVE, see MOUNT below). However, the TYPE=TRANSIT job waits until it is instructed by the FDRMOVE job to move a volume to an available transit station, or move it back to its original device.

In order to separate the messages from these swaps, messages are written to SYSPRIN*x* DD statements ("*x*" is 1-9, 0, and A-V in that order, depending on `MAXTASKS=`*`nn`*). The SYSPRIN*x* DD statements are dynamically allocated as "SYSOUT=*". At the termination of each swap subtask, most of its messages are also written to SYSPRINT so that all FDRPAS messages are in one place.

Default: 5. Do not override the `MAXTASKS=` operand to a smaller value unless you are concerned about the impact of FDRPAS on your production response time (this is rarely a problem). You can override the MAXTASKS= operand to a larger value to move more volumes to the transit stations concurrently.

TRANSITRETURN=

NO – FDRPAS leaves the DASD volume on the transit station device, even after it is no longer needed by FASTMOVE. If data sets remain on this volume, they can benefit from the performance of the new subsystem. This option can be set dynamically with a `MODIFY (F)` command. See "MODIFY Status Command" in Section 325.10.

YES – When the FDRMOVE **"**FASTMOVE**"** job indicates that it is finished with a given DASD volume, the FDRPAS job moves the volume back to its original device. Since FASTMOVE usually moves all or most of the data sets off the volume, this swap may be very quick.

If you intend to stop the FDRMOVE job before it finishes processing all data sets and restart it later; you may want to specify `TRANSITRETURN=NO` so the input volumes are already in the new subsystem

Default: YES.

TRANSIT JOB EXCLUDE STATEMENT

The EXCLUDE statement (see Section 310.4 "SWAP Task EXCLUDE Statement") is accepted and may be required when the source DASD volumes are attached to non-z/OS systems (such as z/VM, VSE, or Linux) or inactive systems. Please read that section to understand when it must be used.

TRANSIT JOB EXCLUDE STATEMENT SYNTAX

The syntax of the EXCLUDE statement for the FDRPAS transit job is:

```
EXCLUDE|    CPUID=cpuid
X
```

TRANSIT JOB MOUNT STATEMENT

The MOUNT statement defines the online source volumes that can be moved to a transit station, and the transit station device addresses to which they can be moved.

TRANSIT JOB MOUNT STATEMENT SYNTAX

The syntax of the MOUNT statement for the FDRPAS transit job is:

```
MOUNT       VOL=&&&&&&|                  ,SWAPUNIT=(uuu1[,uuu2,…])
                 volser
```

TRANSIT JOB MOUNT STATEMENT OPERANDS

VOL=

Specifies the volume serial of an online volume that may be swapped to a transit station device. You have two options:

- If you are going to let FASTMOVE automatically submit the FDRPAS job, you must specify a mask `VOL=&&&&&&`. In this case you must provide only a single MOUNT statement. FDRMOVE replicates this MOUNT statement and all of its associated transit stations (`SWAPUNIT=`) for each DASD volume FASTMOVE determines must be moved. Since there is one replicated MOUNT for each source volume, the `MAXCARDS=` operand must be set if more than 250 volumes are selected.
- If you submit the FDRPAS TYPE=TRANSIT job manually, you can specify a single volume serial, `VOL=volser`. You need to code multiple MOUNT statements, to specify the transit stations for use with each source volume. The transit station list (`SWAPUNIT=`) could be the same for each source volume, or you can provide varying lists of transit stations for each volume. It is your responsibility to include a MOUNT for every source volume that FASTMOVE needs.

SWAPUNIT=

***uuuu* –** Specifies the z/OS device addresses of the offline transit station devices to be associated with a given source volume. The address can be specified as a 4-digit (hex) z/OS device address, or it can be specified as 1, 2, or 3 digits with a trailing asterisk(*); in this case all offline z/OS DASD addresses starting with the prefix specified are used as transit stations. To provide multiple devices or ranges of devices (by prefix), specify them in parentheses, separated by commas, but you cannot specify more than 255 subparameters (device addresses or address prefix ranges) within the parentheses. If you use address ranges, the total number of transit stations can be up to 8190. Only devices that are offline and accessible at the time the monitor starts are used as transit stations; if a device is placed offline or becomes accessible after the SWAP task starts, it is not used.

NOTE: At least two offline units must be specified on the SWAPUNIT= operand unless TRANSITRETURN=NO is also specified.

The transit station devices must be as large as or larger than the input volumes specified by VOL=.

FASTMOVE works best when the number of offline transit station devices equals or exceeds the number of input volumes. FASTMOVE is able to move all the input volumes to a transit station. Then, when you "bounce" the application that is using the data sets, all of them can be moved at once.

However, it may not be possible for you to provide a sufficient number of transit stations. FASTMOVE still works, but in stages. FASTMOVE moves input volumes to the transit stations until it runs out of stations. The FDRMOVE ISPF panel or the console STATUS command shows you what volumes are in transit stations and what data sets FASTMOVE is waiting for. Now when you bounce the application, FASTMOVE moves a subset of the data sets specified.

Assuming that all selected data sets on that set of input volumes have been moved, they are swapped back to their original devices, and other volumes move to the transit stations. Now, you can bounce the application again to move the data sets from those volumes. This is repeated until all selected data sets have been moved.

For example,

```
MOUNT VOL=&&&&&&,SWAPUNIT=(17C0,17C1,17C2)
          Three offline transit station devices.
MOUNT VOL=&&&&&&,SWAPUNIT=17C*
          Transit station devices in the range 17C0-17CF
MOUNT VOL=&&&&&&,SWAPUNIT=(17*,18*,19A*)
          Transit station devices in the ranges
          1700-17FF, 1800-18FF, and 19A0-19AF.
MOUNT VOL=IN0001,SWAPUNIT=(17*,18*)
          Transit station devices in the ranges
          1700-17FF and 1800-18FF.
```

See the notes above under VOL= for information on how to specify the transit stations. Note that the syntax for SWAPUNIT= in a SWAP TYPE=TRANSIT job is different from that in a normal FDRPAS SWAP TYPE=FULL job.

If you have a list of transit station devices, and you specified VOL=&&&&&&, then the MOUNT statement may be continued onto a maximum 15 lines (16 lines total). For example:

```
MOUNT VOL=&&&&&&,SWAPUNIT=(03A0,03A1,
      03A2,03A3,03A4,03A5,03A6,03A7)
```

The MOUNT statement with all its continuations is replicated for each selected source volume.

FDRPAS MONITOR TASKS FOR FDRPAS TRANSIT JOB

An FDRPAS MONITOR task must be executed on every z/OS LPAR that the source DASD is connected to except the LPAR where the SWAP TYPE=TRASNSIT job runs, even if the volume is offline on that LPAR; however, the MONITOR task does not actually participate on the LPAR where the SWAP task is running. For non-z/OS LPARs, such as z/VM and Linux for System z, see the EXCLUDE statement above.

Each FDRPAS MONITOR task must specify the offline transit station DASD volumes and any source DASD volumes that are offline to any of the LPARs. The MONITOR task does not need to run on the LPAR where the SWAP TYPE=TRANSIT job runs. You may need a unique job name for each MONITOR. The JCL is the same as the FDRPAS job shown above. You must start these MONITOR tasks before submitting the FASTMOVE. Note that if an online source DASD volume is included in the SWAPUNIT list, it is simply ignored.

Note that the MONITOR tasks have minimal overhead and have no impact on any DASD volumes or jobs that are not involved in the SWAP. For DASD volumes being swapped to a transit station, the MONITOR installs a low-overhead I/O intercept on the online volume to monitor for updates, but does very little other I/O to those DASD volumes. The intercepts are removed when the volume has been swapped. Therefore, the MONITOR tasks have almost no impact on the LPARs where they run.

MONITOR tasks use little CPU time, but when they need it, to communicate with the SWAP task, they must be able to get it. Two causes for non-responding systems are:

1. The MONITOR task does not have a sufficiently high priority, so that higher priority jobs can prevent it from getting dispatched.
2. The LPAR is capped, allowing it little or no CPU time.

TRANSIT JOB MONITOR STATEMENT SYNTAX

The syntax of the MONITOR statement is:

```
MONITOR     TYPE=SWAP
            ,DURATION=nnnn
```

TRANSIT JOB MONITOR STATEMENT

This statement initiates an FDRPAS MONITOR task that monitors for FDRPAS SWAP tasks beginning a SWAP TYPE=TRANSIT operation. It must be the first statement in the input; only one MONITOR statement is allowed per execution. For more detail on the operation of the MONITOR and other operands that you may want to use. See Section 310.5 “MONITOR SWAP Statement”.

A MONITOR TYPE=SWAP statement must be followed by exactly one MOUNT statement to identify the offline devices to be monitored. For FASTMOVE, this MONITOR task must monitor all transit station swap units specified in the SWAPUNIT= of the FDRPAS job. You may also include regular FDRPAS target devices for swaps in the same MONITOR task.

TRANSIT JOB MONITOR STATEMENT OPERANDS

TYPE=

Must be specified on the MONITOR statement.

SWAP – Initiates a MONITOR task that monitors offline volumes for a swap operation that is initiated by a SWAP task.

DURATION=

***nnnn* –** Specifies the number of idle minutes that the MONITOR task executes; it does not include time that the MONITOR task is actively participating in a swap. The MONITOR task automatically terminates when it has been idle for a total of this many minutes.

Default: The MONITOR task executes until it is terminated by a console STOP (P) command, canceled, or until all devices that it is monitoring have been swapped or are online.

NOTE: If the FASTMOVE job may execute for a long time, `DURATION=` should be omitted.

MOUNT STATEMENT SYNTAX

The syntax of the MOUNT statement is:

MOUNT	SWAPUNIT=(*uuu1*[,*uuu2*,…])

MOUNT STATEMENT

The monitor MOUNT SWAPUNIT= statement follows the MONITOR TYPE=SWAP statement and specifies an offline target device or set of devices to be monitored to see if one or more of them is selected as the target of a swap by an FDRPAS TYPE=TRANSIT or a regular FDRPAS SWAP task on another LPAR. Only one MOUNT statement can be specified, specifying the devices to be monitored by this MONITOR task.

After a TYPE=TRANSIT swap, the MONITOR task automatically switches to monitoring the original input DASD volume (now offline) so that the volume can be returned to the original DASD volume when requested by FASTMOVE. There is no need to specify the original DASD units. When the volume is swapped back to its original location the MONITOR task again switches to monitor the offline transit station. Therefore, the same MONITOR tasks are used for the swap to the transit station and then back to the original device; there is no need to stop and restart the MONITOR tasks in between.

NOTE: If a MONITOR task is stopped (because of a console STOP (P) command or the DURATION= operand or an error), and there are volumes still in transit stations, you should restart the MONITOR tasks with SWAPUNIT= specifying the original DASD devices of those volumes, in addition to all the transit station devices, so that the MONITOR tasks can successfully swap a volume in and out of a transit station.

MOUNT STATEMENT OPERANDS

SWAPUNIT=

uuuu – Specifies the z/OS device addresses of the target devices to be monitored. The address can be specified as a 4-digit (hex) z/OS device address, or it can be specified as 1, 2, or 3 digits with a trailing asterisk(*); in this case all offline z/OS device addresses starting with the prefix specified are monitored. To monitor multiple devices or ranges of devices, specify them in parentheses, separated by commas, but you cannot specify more than 255 subparameters (device addresses or address ranges) within the parentheses. If you use address ranges, the total number of devices to monitor can be up to 8190. Only devices that are offline and accessible at the time the MONITOR task starts are monitored; if a device is placed offline or becomes accessible after the MONITOR task starts, it is not monitored.

For example,

```
MOUNT SWAPUNIT=17C0
          Monitors one offline device 17C0.
MOUNT SWAPUNIT=17C*
          Monitors offline devices in the range 17C0-17CF.
MOUNT SWAPUNIT=(17*,18*,19A*)
          Monitors offline devices in the ranges 1700-17FF,
          1800-18FF, and 19A0-19AF.
```

325.3 FDRMOVE MOVE FUNCTION WITH EXAMPLES

MOVE FUNCTION OF FDRMOVE

The MOVE function of FDRMOVE moves the selected data sets when each of them becomes inactive, such as when batch jobs using the data sets finish, or TSO users logoff. Normally, regular read/write I/O is used to move the data sets so larger data sets may be unavailable for longer periods. For this reason, MOVE is used for data sets that can tolerate being unavailable for longer periods than the data sets selected for FASTMOVE.

A single FDRMOVE job can process many volumes, but if you wish to reduce the total elapsed time, you can start multiple FDRMOVE jobs, each processing different input volumes.

In many installations, 90% of the data sets are inactive (not in use) or become inactive over some period of time. These data sets typically remain inactive for a long period of time. MOVE automates the process of moving these data sets with little or no impact. If a job needs the data set while it is being moved, it usually waits for it.

SELECT statements identify the data sets to be moved and the new volume(s) to move them onto. You can move:

- Entire volumes (all data sets on selected input volumes). This is especially useful when you want to combine specific small volumes onto one or more larger volumes.
- All data sets on one or more volumes that are below a certain size. Testing has shown that moving small data sets with normal I/O is often faster than using instant replication technology to move them.
- Specified data sets, selected by name and volume serial or SMS storage group or optionally from the catalog.

MOVE identifies all of the input volumes according to your specifications. It then begins moving inactive data sets from each volume, up to eight volumes concurrently per FDRMOVE job. When it has completed this first pass of all volumes, it begins waiting for active data sets to become available.

MOVE attempts to move each selected data set, but if the data set is active (a SYSDSN enqueue exists for the data set), and then it is bypassed. MOVE periodically tests each enqueued data set to see if it has become free, and if so moves it. The MOVE job can run for hours or days if needed, selecting data sets as they become free, until all selected data sets have been moved, or you terminate the FDRMOVE step. The MOVE function usually uses normal read/write I/O to move the data sets, but it may also use instant replication functions like FlashCopy and EMCSNAP if the source and target DASD volumes are in the same subsystem.

Moved data sets are allocated and cataloged on the output volume, and scratched from the input volume. FDRMOVE holds an exclusive SYSDSN enqueue on each data set while it is actually moving it, so each data set is not available while it is being moved, but is immediately available as soon as it is moved. FDRMOVE offers a Dynamic Allocation exit to prevent dynamic allocation from failing if a job needs a data set while it is being moved. With the FDRMOVE Dynamic Allocation exit installed, failed dynamic allocations are monitored and if it is found that the failure is caused by a data set that is actively being moved by FDRMOVE, this dynamic allocation is intercepted and delayed. This delay is comparable to a recall of an archived data set when a task requires a data set that was archived. See Section 380.18 “Dynamic Exit Installation Procedure” for details on installing the exit and the options available for this exit. This exit works for both MOVE and FASTMOVE processes, but since the FASTMOVE process is moving data sets with “Instant” technology, the “window” of contention is less for FASTMOVE.

The MOVE job consists of a simple PGM=FDRMOVE step, invoking the MOVE function. Unlike FASTMOVE, no other jobs are required. You may also be able to improve performance (more volumes processed concurrently) by running multiple MOVE jobs selecting different subsets of the data to be moved, they can use the same target volumes.

You can monitor the progress of MOVE with the console “F *movejob*,STATUS” command or the FDRMOVE ISPF panel. See “MODIFY Status Command” in Section 325.10.

MOVE EXAMPLES

Here are a number of examples of the FDRMOVE MOVE function. All examples in this section are found in the JCL library installed with FDRPAS. The member names are PA32503*x*.

CONSOLIDATE DASD EXAMPLE

This is an example of a MOVE to consolidate (60) 3390-9 input volumes to (22) 3390-27 output volumes.

The default of `MAXTASKS=8` allows eight input volumes to be processed concurrently using internal subtasks. All data sets on the input volumes are moved immediately if they are not active (SYSDSN enqueue). Active data sets are retried at 15-second intervals; as soon as they become inactive, they are moved.

FDRMOVE terminates automatically if all selected data sets have been moved. However, if some data sets never become inactive, `STOPINACTIVE=480` causes FDRMOVE to automatically terminate if it has been idle (not moving data sets) for a cumulative time of 480 minutes (8 hours).

```
//FDRMOVE  EXEC PGM=FDRMOVE,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//FDRSUMM  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  MOVE     TYPE=DSF,DISABLENEW=YES,STOPINACTIVE=480
  SELECT   ALLDSN,VOL=IN0001,NVOL=(OUT001,OUT002,…,OUT022)
  SELECT   ALLDSN,VOL=IN0002,NVOL=(OUT001,OUT002,…,OUT022)
  SELECT   ALLDSN,VOL=IN0003,NVOL=(OUT001,OUT002,…,OUT022)
  …
  SELECT   ALLDSN,VOL=IN0059,NVOL=(OUT001,OUT002,…,OUT022)
  SELECT   ALLDSN,VOL=IN0060,NVOL=(OUT001,OUT002,…,OUT022)
/*
```

The `NVOL=` list specifies (22) 3390-27 volumes. FDRMOVE round-robins (rotate) the volumes in the NVOL list for each ALLDSN statement. Data sets from IN0001 go to OUT001, IN0002 goes to OUT002, and so on, and IN0023 goes to OUT001, IN0024 goes to OUT002, and so on. Although the contents of the input volumes would usually fit on 20 output volumes, two extra target volumes are provided to handle very full volumes and activity on the target volumes by other jobs.

CONSOLIDATE STORAGE GROUP EXAMPLE

This is an example of a MOVE to consolidate the 3390-9 input volumes in an SMS storage group to 3390-27 output volumes in the same storage group, but in a different control unit. The user has previously added all new volumes in the target control unit to the same SMS storage group.

FDRMOVE terminates automatically if all selected data sets have been moved. DISABLENEW=YES sets all input volumes to DISNEW status so that new allocations go to the output control unit.

```
//FDRMOVE  EXEC PGM=FDRMOVE,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//FDRSUMM  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  MOVE     TYPE=DSF,DISABLENEW=YES
  SELECT   ALLDSN,CUID=12345,STORGRP=DB2APPL,
           NEWCUID=54321,NEWSTORGRP=DB2APPL
/*
```

The CUID=*xxxxx* *operand* specifies the serial number of the source control unit and the NEWCUID=*sssss* *operand* specifies the serial number of the target control unit. FDRMOVE round-robins (rotate) the volumes in the target control unit for each of the ALLDSN statements that are internally generated.

NOTE: If a different SMS storage group is specified for NEWSTORGRP=, change the SELECT statement above to:

```
SELECT ALLDSN,STORGRP=DB2APPL,NEWSTORGRP=DB2APPL2
```

LIMIT BY SIZE EXAMPLE

If you are concerned that large data sets may be unavailable for many minutes while they are being moved, this example shows how to limit MOVE to data sets of 1500 tracks (100 cylinders) or less. A FASTMOVE step can be used to quickly move the larger data sets that remain. See the notes in the first example about NVOL=. If any of the data sets being moved may be dynamically allocated by a job during the move process, it is recommended installing the FDRMOVE Dynamic Allocation exit.

```
//FDRMOVE  EXEC PGM=FDRMOVE,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//FDRSUMM  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  MOVE     TYPE=DSF,DISABLENEW=YES
  EXCLUDE  ALLDSN,SIZE=1500  EXCLUDE DATA SETS > 100 CYLINDERS
  SELECT   ALLDSN,VOL=IN0001,NVOL=(OUT001,OUT002,OUT003)
  SELECT   ALLDSN,VOL=IN0002,NVOL=(OUT001,OUT002,OUT003)
/*
```

SELECT FROM CATALOG EXAMPLE

This is an example of a MOVE that selects data sets on specific input volumes by selecting them from the system catalogs. The cataloged data sets selected reside on (30) 3390-3 volumes. They are copied to (11) 3390-9 volumes (this assumes that the input volumes are fairly full with DB2 data; if not, you may be able to use fewer output volumes). FDRMOVE automatically round-robins the NVOL= list for each source volume selected.

If you are unsure how many input volumes are involved, change MOVE to SIMMOVE. The FDR346 message shows how many volumes the CATDSN= selected.

```
//FDRMOVE  EXEC PGM=FDRMOVE,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//FDRSUMM  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  MOVE     TYPE=DSF
  SELECT   CATDSN=DB2.**,NVOL=(OUT001,…,OUT011)
/*
```

SAMPLE MOVE OUTPUT

These are extracts from the output of an FDRMOVE MOVE step. Data sets that are not active (no SYSDSN enqueue exists for the data set) are moved immediately. The data sets that get the enqueue failure are remembered; they are retried at intervals and as soon as they are no longer active, they are moved.

The inactive data sets are immediately moved to their new volume.

```
FDR001  FDR PLUG AND SWAP DATA SET - FDRMOVE  VER. 5.4/xxR  - INNOVATION DATA PROCESSING
FDR303  CARD IMAGE --  MOVE TYPE=DSF
FDR333  CARD IMAGE -- EXCLUDE TEMP                                          UNMOVABLE TABLE
FDR303  CARD IMAGE --  S CATDSN=DB2.**,NVOL=(OUT001,OUT002,OUT003,OUT004 ,…)
FDR346  FILTER SELECTED      60 VOLUMES IN     0.43 SECS
FDR256  --- REPORT FROM SYSPRIN1 ---
FDR007  STARTING TIME OF DATA SET MOVE    -- 11.26.40 -- IN=D#IN0001
FDR314   FDR SCRATCHED    DSN=DB2.MOVE039.VIN0001.PS.ONETRK.A0005                    FROM VOL=IN0001
FDR311   FDR MOVED    DSN=DB2.MOVE039.VIN0001.PS.ONETRK.A0005         ALLOCATED CATALOGED
FDR311               ON VOLSER=OUT001  UNIT=3390-27 (IN0001)
FDR314   FDR SCRATCHED    DSN=DB2.MOVE039.VIN0001.PS.ONETRK.A0004                    FROM VOL=IN0001
FDR311   FDR MOVED    DSN=DB2.MOVE039.VIN0001.PS.ONETRK.A0004         ALLOCATED CATALOGED
FDR311               ON VOLSER=OUT001  UNIT=3390-27 (IN0001)
FDR007  ENDING   TIME OF DATA SET MOVE    -- 11.26.40 -- IN=D#IN0001
```

A console STATUS request or ISPF panel (see Section 325.10 "FDRMOVE Console Commands and ISPF Interface") shows the active data sets. FDRMOVE tests the enqueue status of each data set at intervals.

```
FDR184  -- MOVE STATUS REPORT --
FDR184  VOL=IN0002 AWAITING MOVE FROM IBM76421 TO EMC00547     356 DATASETS TO MOVE
FDR184         |---FOR ACTIVE DSN=DB2.MOVE039.MV.KSDS.CLUSTER                  1 JOBS(DB2PROD)
FDR184         |---FOR ACTIVE DSN=DB2.MOVE039.VIN0002.KSDS.A0001.CLUSTER       2 JOBS(MOVE039Q,MOVE039E)
FDR184         |---FOR ACTIVE DSN=DB2.MOVE039.VIN0002.PO.A0001                 2 JOBS(MOVE039Q,MOVE039E)
FDR184         |---FOR ACTIVE DSN=DB2.MOVE039.VIN0002.PS.A0001                 2 JOBS(MOVE039Q,MOVE039E)
FDR184         |---FOR ACTIVE DSN=DB2.MOVE039.VIN0002.PS.A0002                 1 JOBS(DB2PROD)
FDR184         |---FOR ACTIVE DSN=DB2.MOVE039.VIN0002.PS.A0004                 1 JOBS(DB2PROD)
FDR184  VOL=IN0003 AWAITING MOVE FROM IBM76421 TO EMC00547       4 DATASETS TO MOVE
FDR184         |---FOR ACTIVE DSN=DB2.MOVE039.VIN0003.KSDS.A0002.CLUSTER       1 JOBS(DB2PROD)
FDR184         |---FOR ACTIVE DSN=DB2.MOVE039.VIN0003.PO.A0002                 1 JOBS(DB2PROD)
FDR184         |---FOR ACTIVE DSN=DB2.MOVE039.VIN0003.PS.A0001                 2 JOBS(MOVE039Q,MOVE039E)
FDR184         |---FOR ACTIVE DSN=DB2.MOVE039.VIN0003.PS.A0002                 1 JOBS(DB2PROD)
FDR184  FDR MOVED       615 DATASETS/COMPONENTS SUCCESSFULLY
FDR184  MOVE STATUS     650 ACTIVE DATASETS    0 AWAITING TRANSIT    0 MOVING TO TRANSIT    0 IN TRANSIT STATION
FDR265  MODIFY COMMAND COMPLETED - STATUS
```

When data sets become free, they are moved.

```
FDR314   FDR SCRATCHED    DSN=DB2.MOVE039.VIN0002.PS.A0004                    FROM VOL=IN0002
FDR311   FDR MOVED    DSN=DB2.MOVE039.VIN0002.PS.A0004         ALLOCATED CATALOGED
FDR311               ON VOLSER=SY3100  UNIT=3390-27 (IN0002)
FDR314   FDR SCRATCHED    DSN=DB2.MOVE039.VIN0002.PS.A0002                    FROM VOL=IN0002
FDR311   FDR MOVED    DSN=DB2.MOVE039.VIN0002.PS.A0002         ALLOCATED CATALOGED
FDR311               ON VOLSER=SY3100  UNIT=3390-27 (IN0002)
FDR007  ENDING   TIME OF DATA SET MOVE    -- 11.28.08 -- IN=D#SH20C0
```

When FDRMOVE has moved all selected data sets, it terminates.

```
FDR184  FDR MOVED       129 DATASETS/COMPONENTS SUCCESSFULLY
FDR999   FDR SUCCESSFULLY COMPLETED
```

MOVE WITHIN THE SAME STORAGE GROUP EXAMPLE

Move all data sets from smaller devices to 3390-27 in the same SMS storage group. Since the data set's source and target are in the same SMS storage group, the STORGRP= and NEWSTORGRP= operands cannot both be specified. In this case, FDREPORT is used to obtain the volume information and generate VOL= and NVOL= operands from the SMS storage group that can be used by FDRMOVE.

```
//FDREPORT EXEC PGM=FDREPORT,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdr.loadlib
//SYSPRINT DD SYSOUT=*
//ABRMAP   DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//FDRLIB   DD *
)PREFIX
  SELECT   ALLDSN,VOL=@VLVOLSER#,
)ENDPREFIX
)CONTVALUES |  |,|,|
            NVOL=(<VLVOLSER>)
/*
//ALTMASK  DD UNIT=SYSALLDA,SPACE=(TRK,(10,10))
//SYSPUNCH DD SYSOUT=*
//SYSIN    DD *
  DEFAULT  SUMDEV=UNIQUE
  XSELECT  VLSMSSTG=PROD*
  PRINT    DATATYPE=VOLDATA
  CANCEL   SELECT
  XSELECT  VLSMSSTG=PROD*,VLDEVTYP=3390-27
  PUNCH    ECHO
  PRINT    DATATYPE=VOLDATA,RPTYPE=SELPCH,PCHDD=ALTMASK
  CANCEL   SELECT
  DEFAULT  FIELDPREFIX=@,FIELDSUFFIX=#
  XEXCLUDE VLDEVTYP=3390-27
  XSELECT  VLSMSSTG=PROD*,VLDEVTYP=(3390-2,3390-3)
  PUNCH    FDRLIB=ALTMASK,ECHO
  PRINT    DATATYPE=VOLDATA,RPTYPE=SELPCH
/*
```

This is an example of the output that would be generated on the SYSPUNCH by this FDREPORT job:

```
SELECT ALLDSN,VOL=SYSLB6,
NVOL=(VTS017,VTS018,VTS019,VTS020,VTS021,VTS022,VTS023,VTS024,VTS025,
 VTS026,VTS027,VTS028,VTS029,VTS030,VTS031,VTS032,SYTS45,SYTS46,SYTS47,
 SYTS48)
SELECT ALLDSN,VOL=SYSLB2,
NVOL=(VTS017,VTS018,VTS019,VTS020,VTS021,VTS022,VTS023,VTS024,VTS025,
 VTS026,VTS027,VTS028,VTS029,VTS030,VTS031,VTS032,SYTS45,SYTS46,SYTS47,
 SYTS48)
SELECT ALLDSN,VOL=IDPLB6,
NVOL=(VTS017,VTS018,VTS019,VTS020,VTS021,VTS022,VTS023,VTS024,VTS025,
 VTS026,VTS027,VTS028,VTS029,VTS030,VTS031,VTS032,SYTS45,SYTS46,SYTS47,
 SYTS48)
SELECT ALLDSN,VOL=SYSLB3,
NVOL=(VTS017,VTS018,VTS019,VTS020,VTS021,VTS022,VTS023,VTS024,VTS025,
 VTS026,VTS027,VTS028,VTS029,VTS030,VTS031,VTS032,SYTS45,SYTS46,SYTS47,
 SYTS48)
…
```

325.4 FDRMOVE FASTMOVE FUNCTION WITH EXAMPLES

FASTMOVE FUNCTION OF FDRMOVE

The FASTMOVE function of FDRMOVE uses instant replication functions (FlashCopy, SNAP, and EMCSNAP) to quickly move data sets that can tolerate only very brief outages, even when the source and target DASD are in different DASD subsystems.

Data sets can be selected from the catalog, with a control statement such as:

```
SELECT CATDSN=ABC.**,NVOL=XYZ*
```

Usually this is used to select the data sets belonging to a single application. Or you can select all data sets on an input volume with a control statement such as:

```
SELECT ALLDSN,VOL=IN0001,NVOL=XYZ*
```

For data sets that are active for a long time, such as database, CICS or 24x7 applications, it is necessary to schedule a time to close and deallocate the files or shutdown the application for a brief time, while FASTMOVE uses FlashCopy, SNAP, or EMCSNAP to quickly move the data sets to the new volumes. This is known as "bouncing" the application. FASTMOVE can move large amounts of data very quickly (typically up to 1TB per minute) so the "bounce" time is very small.

Moved data sets are allocated and cataloged on the output volume, and scratched from the input volume. FDRMOVE holds an exclusive SYSDSN enqueue on each data set while it is actually moving it, so each is not available while it is being moved, but is immediately available as soon as it is moved.

To use FASTMOVE:

- You must provide a set of unused offline DASD devices in the new subsystem (where the target DASD reside). These are known as "transit station DASD volumes." The transit station DASD volumes can be the same size as the source DASD (or larger), but you must usually provide a number of transit stations equal to the number of source DASD volumes containing data sets belonging to the application (see "Transit Stations" in Section 325.11 for notes on this).
- If you have multiple LPARs (and almost everyone does), you must start FDRPAS MONITOR tasks on all LPARs (it's not necessary on the one FASTMOVE is running on) before submitting the FASTMOVE. These MONITOR tasks must monitor the offline transit station DASD volumes. See Section 310.5 "MONITOR SWAP Statement", Section 310.6 "MONITOR SWAP MOUNT Statement", and Section 310.23 "MONITOR SWAP Examples" for details on setting up the MONITOR tasks. Note that a MONITOR task can participate in regular FDRPAS swaps as well as FASTMOVE; simply specify all the swap target addresses and transit station addresses.
- Start the FASTMOVE job, using a simple PGM=FDRMOVE step invoking the FASTMOVE function and specifying the data sets or volumes to be moved.
- FASTMOVE identifies the volumes containing the input data sets and submit a special FDRPAS job to move the input volumes to the transit stations. If FASTMOVE finds that a data set is already on a volume in the `NVOL=` list, it is not moved.
- Once the FDRPAS job starts, FASTMOVE signals the FDRPAS jobs to swap the online source volumes to one of the transit station DASD volumes.
- Once a volume reaches a transit station, inactive data sets are immediately moved. Data sets that are active (SYSDSN enqueue) are queued and re-tested at frequent intervals. As soon as those data sets are no longer active, they are moved. Because the transit station devices are in the same DASD subsystem as the target DASD volumes, instant replication (FlashCopy, SNAP, or EMCSNAP) is used to quickly move the data sets.
- When all source DASD volumes are in the transit station and data sets are still in use, the application that is using them can be bounced. The console STATUS command or the ISPF interface can be used to check volume status and identify the data sets FASTMOVE still needs (see Section 325.10 "FDRMOVE Console Commands and ISPF Interface").

- When the selected data sets become inactive (dequeued), FASTMOVE uses instant replication technology to rapidly move them to the pool of target DASD volumes. An exclusive SYSDSN enqueue is held while the data set is being moved, which may allow the application to be immediately restarted (it waits for FDRMOVE to dequeue the data set). Data sets that are still active are remembered, and are checked at intervals; as soon as they become inactive they are moved.
- Once FASTMOVE has moved all data sets selected on the source volumes (or it is instructed to stop), it again signals FDRPAS to swap the source volumes back to the original source DASD volumes. Since there are usually no or far fewer data sets left on the DASD, this swap is very fast.
- If you must terminate FDRMOVE before it completes all data sets, you have the option of leaving the volumes in the transit stations (`F jobname TRANSITRETURN=NO`). When you restart FDRMOVE it recognizes that they are already in the new subsystem. As long as FASTMOVE runs on the same LPAR and an IPL has not occurred, it recognizes that the DASD volumes are transit stations; otherwise, it does not move the volumes back to their original DASD volumes.

Normal I/O is used if the total tracks to be moved from one input volume are 15 or less.

A given FASTMOVE job should only move data sets to a single target DASD subsystem, although the input data sets may be on multiple subsystems. If you need to move to multiple subsystems, reserve transit stations in each subsystem and run multiple FASTMOVE jobs. You may also be able to improve performance (more volumes processed concurrently) by running multiple FASTMOVE jobs selecting different subsets of the data to be moved, they can use the same target volumes and transit stations if you like.

FASTMOVE EXAMPLES

Here are a number of examples of the FDRMOVE FASTMOVE function. All the examples in this section are found in the JCL library installed with FDRMOVE. The member names are PA32504*x*.

MOVE CATALOGED DATA SETS EXAMPLE

This is an example of a FASTMOVE that selects data sets from the system catalogs. The input volumes are moved to the transit station units, in the same control unit as the output volumes, so that fast replication (such as FlashCopy and EMCSNAP) can be used. Inactive data sets are moved once a DASD volume is in a transit station. For active data sets, FASTMOVE waits for the data set to become inactive (such as shutting down its application) and quickly and automatically moves the data set so that the application can be restarted in only a few minutes.

In this example, FASTMOVE selects certain cataloged data sets starting with ABC and XYZ with PAYROLL anywhere else in the name.

The output volumes, OUT001 to OUT009, are in the target subsystem. As described in "Output Volume Selection for FDRMOVE" in Section 325.2, FDRMOVE rotates the output volumes, so that the data sets from different input volumes go to different output volumes.

```
//FDRMOVE  EXEC PGM=FDRMOVE,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//FDRSUMM  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  FASTMOVE TYPE=DSF
  SELECT   CATDSN=ABC.**,NVOL=OUT00*
  SELECT   CATDSN=XYZ.**PAYROLL**,NVOL=OUT00*
/*
//PASJOB   DD DATA,DLM=$$
//TRANSIT1 JOB (accounting info),FDRMOVE,CLASS=c,
//         MSGCLASS=m  <-- add any other required JOB parameters
//PASTRANS EXEC PGM=FDRPAS,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//FDRSUMM  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  SWAP     TYPE=TRANSIT
  MOUNT    VOL=&&&&&&,SWAPUNIT=32*
$$
```

If FASTMOVE finds that one or more source volumes must be moved to a transit station in the target control unit, it submits the FDRPAS job pointed to by the PASJOB DD statement. In this example, it is an in-stream job but it could also point to a member of a PDS/PDSE. `DATA` with `DLM=` allows the transit job JCL and control statements to be an input data set to FASTMOVE.

The FDRPAS transit station job is called TRANSIT1 in this example. If you submit multiple FASTMOVE jobs, each transit station job must have a unique job name. The MOUNT statement is internally replicated for each volume that may need to be moved to a transit station, substituting the actual volume serial for `&&&&&&`. In this example, the transit stations are all offline DASD devices in the range of 3200 to 32FF.

Normal FDRPAS "MONITOR TYPE=SWAP" jobs must already be running on the other systems in the complex, specifying the same transit station DASD volumes (`SWAPUNIT=`). See Section 325.5 "FDRMOVE Target Volume Initialization" for details.

CONSOLIDATE STORAGE GROUP EXAMPLE

This is an example of a FASTMOVE to consolidate the input volumes in an SMS storage group to output volumes in a different storage group. SMS Automatic Class Selection (ACS) routines must be updated to select the new storage group for new allocations.

FDRMOVE terminates automatically if all selected data sets have been moved. DISABLENEW=YES sets all input volumes to DISNEW status so that new allocations go to the output control unit.

```
//FDRMOVE  EXEC PGM=FDRMOVE,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//FDRSUMM  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  FASTMOVE TYPE=DSF,DISABLENEW=YES
  SELECT   ALLDSN,STORGRP=CICSAPPL,NEWSTORGRP=CICSNEW
/*
//PASJOB   DD  DISP=SHR,DSN=SYSPROG.CNTL(PASJOB)
```

MOVE ENTIRE VOLUMES EXAMPLE

This is an example of a FASTMOVE that selects all data sets on specific input volumes. Otherwise, it is similar to the previous example. The PASJOB DD statement points to a PDS member, which must contain the FDRPAS job stream. DISABLENEW=YES sets the input volumes to DISNEW (SMS) or PRIVATE (non SMS-managed) so that the volumes are no longer used for new data sets; remove it if you do not plan to remove the input subsystem.

```
//FDRMOVE  EXEC PGM=FDRMOVE,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//FDRSUMM  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  FASTMOVE TYPE=DSF,DISABLENEW=YES
  SELECT   ALLDSN,VOL=IN0001,NVOL=OUT00*
  SELECT   ALLDSN,VOL=IN0002,NVOL=OUT00*
  SELECT   ALLDSN,VOL=IN0003,NVOL=OUT00*
  SELECT   ALLDSN,VOL=IN0004,NVOL=OUT00*
  SELECT   ALLDSN,VOL=IN0005,NVOL=OUT00*
  SELECT   ALLDSN,VOL=PROD0*,NVOL=PRODN*
/*
//PASJOB   DD DISP=SHR,DSN=SYSPROG.CNTL(PASJOB)
```

SIMMOVE

SIMMOVE simulates a FASTMOVE step. It identifies all of the selected input data sets and the volumes they reside on. If the optional PASJOB DD statement is present, it submits the FDRPAS TYPE=TRANSIT job to be sure that it begins executing, and then immediately stops it.

If used with the optional VTOCEMPTY=CHECK operand, it verifies that all selected data sets are in fact moved. For SELECT ALLDSN, it verifies that the input volumes are empty except for the VTOC, VTOCIX, and VVDS.

This example validates a complete FASTMOVE step, from the previous example.

```
//FDRMOVE  EXEC PGM=FDRMOVE,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//FDRSUMM  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  SIMMOVE  TYPE=DSF,DISABLENEW=YES
  SELECT   ALLDSN,VOL=IN0001,NVOL=OUT00*
  SELECT   ALLDSN,VOL=IN0002,NVOL=OUT00*
  SELECT   ALLDSN,VOL=IN0003,NVOL=OUT00*
  SELECT   ALLDSN,VOL=IN0004,NVOL=OUT00*
  SELECT   ALLDSN,VOL=IN0005,NVOL=OUT00*
  SELECT   ALLDSN,VOL=PROD0*,NVOL=PRODN*
/*
//PASJOB   DD DISP=SHR,DSN=SYSPROG.CNTL(PASJOB)
```

This example with VTOCEMPTY=CHECK verifies that the input volumes have no data sets remaining. If you intend to move all data off the old DASD subsystem, this helps you to know when you are ready to disconnect the subsystem.

```
//FDRMOVE  EXEC PGM=FDRMOVE,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//FDRSUMM  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  SIMMOVE  TYPE=DSF,VTOCEMPTY=CHECK
  SELECT   ALLDSN,VOL=IN0001
  SELECT   ALLDSN,VOL=IN0002
  SELECT   ALLDSN,VOL=IN0003
  SELECT   ALLDSN,VOL=IN0004
  SELECT   ALLDSN,VOL=IN0005
  SELECT   ALLDSN,VOL=PROD0*
/*
```

SAMPLE FASTMOVE OUTPUT

The messages from a FASTMOVE job are similar to those shown for MOVE. See Section 325.3 "FDRMOVE MOVE Function with Examples", except that you see messages relating to transit stations, such as:

```
FDR182  FDR REQUESTS THAT VOL=IN0001 MOVE   TO THE TRANSIT STATION EMC00547
FDR183  FDR SUBMITTED FDRPAS TRANSIT JOBNAME=MOVE013B
```

In addition, the FDR311 message includes the word INSTANT to indicate that fast replication technology, like FlashCopy and EMCSNAP, has been used. For data sets 15 tracks or less, normal I/O is used and INSTANT does not appear.

```
FDR314   FDR SCRATCHED    DSN=DB2.MOVE001.PS.VSY3009.A0002              AND UNCATALOGED FROM VOL=IN0001
FDR311   FDR MOVED     DSN=DB2.MOVE001.PS.VSY3009.A0002                ALLOCATED CATALOGED INSTANT
FDR311                 ON VOLSER=OUT001  UNIT=3390     (IN0001)
```

SAMPLE SIMMOVE OUTPUT

The messages from a SIMMOVE job show you all of the selected data sets (and their current enqueue status), all of the input volumes, and transit station information.

A SIMMOVE with `VTOCEMPTY=CHECK` verifies that all data sets have been moved. Here is an example with `SELECT ALLDSN`, so that it verifies the input volumes are empty. If any volume still contained data sets, it would list the data sets:

```
FDR303  CARD IMAGE --  SIMMOVE TYPE=DSF,VTOCEMPTY=CHECK
FDR333  CARD IMAGE -- EXCLUDE DSN=SYS1.VTOCIX.**                    UNMOVABLE TABLE
FDR333  CARD IMAGE -- EXCLUDE DSN=SYS1.VVDS.**                      UNMOVABLE TABLE
FDR303  CARD IMAGE --  S ALLDSN,VOL=IN0001
FDR303  CARD IMAGE --  S ALLDSN,VOL=IN0002
FDR303  CARD IMAGE --  S ALLDSN,VOL=IN0003
FDR303  CARD IMAGE --  S ALLDSN,VOL=IN0004
FDR184   VTOC IS EMPTY ON ALL VOLUMES
FDR999   FDR SUCCESSFULLY COMPLETED
```

325.5 FDRMOVE TARGET VOLUME INITIALIZATION

The target volumes must be prepared. INNOVATION DATA PROCESSING recommends that you create brand-new target volumes, initializing them with appropriate sized VTOCs for the number of data sets you expect to place on the volumes.

FDRMOVE includes a volume initialization utility, FDRINITV, that can easily initialize many target volumes with simple control statements. Although volume initialization can be done with the IBM ICKDSF utility, it requires coding an INIT statement for each volume to be processed. FDRINITV is much more convenient.

Here is a sample FDRINITV job to initialize many 3390-27 DASD volumes very quickly.

```
//FDRINITV EXEC PGM=FDRINITV,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//FDRSUMM  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  INITVTOC TYPE=FULL,VARYON=AFTER,CHECKTARGET=YES
  MOUNT    INITUNIT=21C*,CHANGEVOL=DB////,
           VTOCLOC=15,VTOCSIZE=750
/*
```

FDRINITV is based on the EMPTYVTOC function of FDRERASE, except that the main statement is `INITVTOC` instead of `EMPTYVTOC`. For more information on the options available with FDRINITV, please read the information on "EMPTYVTOC Statement" in Section 330.4.

In the FDRINITV example above, the DASD volumes must be offline before FDRINITV starts but `VARYON=AFTER` varies them online after initialization. DASD volumes 21C0-21CF are initialized as non SMS-managed volumes, with volume serials from DB0000 to DB0015. `VTOCLOC=15` starts the VTOC at cylinder 1 and VTOCSIZE=750 causes it to occupy 50 full cylinders (750 tracks). FDRINITV calculates an appropriate size for the VTOCIX based on the VTOC size. `CHECKTARGET=YES` verifies that the volumes to be initialized do not contain any data sets.

To initialize SMS-managed volumes, add the STORAGEGROUP operand to the MOUNT statement, for example,

```
MOUNT INITUNIT=21C*,CHANGEVOL=DB////,VTOCLOC=15,VTOCSIZE=750,
      STORAGEGROUP
```

After initializing the SMS-managed volumes, you must use ISMF to add those volumes to a storage group. INNOVATION DATA PROCESSING strongly recommends that this be the same storage group used by the input data sets to be moved with FDRMOVE.

There are 50 Data Set Control Blocks (DSCBs) per track on a 3390 VTOC; each DSCB describes one data set or VSAM component. Extra DSCBs are required for data sets over three extents. Therefore, a 30 track (two cylinder) VTOC holds about 400 data sets/components. A 750 track VTOC holds over 35,000 data sets. The VTOCIX needs to be only fraction of the size of the VTOC. Please read an appendix in the IBM *Device Support Facilities (ICKDSF) User's Guide and Reference (GC35-0033)* for more information on VTOC and VTOCIX sizes.

If you are initializing a smaller DASD (for example, 3390-9) or a larger DASD (for example, 3390-54) you can adjust the VTOC size appropriately. If you expect the volume to hold only a few large data sets, you may want to make the VTOC smaller.

It is possible to use FDRPAS to move some volumes to the target DASD volumes and then use FDRMOVE to add data sets from other volumes to fill up the free space on the larger DASD. However, the VTOCs on the volumes moved with FDRPAS must be large enough to accommodate the extra data sets. The example (see Section 325.9 "Consolidate Volumes with FDRPAS and FASTMOVE") shows how to consolidate volumes with FDRPAS and FDRMOVE with notes on checking VTOC sizes and

expanding VTOCs. FDRPAS is recommended only for moving volumes containing data sets that FDRMOVE cannot handle, such as catalogs (see Section 325.11 "FDRMOVE Special Considerations").

325.6 FDRMOVE GENERAL PROCEDURE

Usually, you want to execute the MOVE function of FDRMOVE to move data sets that are inactive or that become inactive while MOVE is running. This typically includes data sets used by batch jobs or TSO users. You can leave MOVE running for hours or days, and it automatically recognizes selected data sets that have become free so that they can be moved with normal I/O. You may want to limit the size of data sets selected by MOVE since small data sets can often be moved faster with normal I/O than with instant replication.

The data sets remaining after MOVE are those in use by long-running applications and larger data sets. FASTMOVE is used to move those data sets with instant replication very quickly, so that the time that the applications must be disrupted are very short.

MOVE and FASTMOVE jobs terminate when:

- All selected data sets have been moved
- You enter a console `STOP (P)` command for the job
- The time limit specified by the `STOPAFTER=` and/or `STOPINACTIVE=` operands is reached.

When FASTMOVE terminates, it also terminates the FDRPAS job it submitted. FDRPAS swaps all source volumes back to their original devices before it ends. They probably contain only a few data sets (or perhaps no data sets) so this swap should be very fast.

MOVE and FASTMOVE are described in more detail in:

- Section 325.3 “FDRMOVE MOVE Function with Examples”
- Section 325.4 “FDRMOVE FASTMOVE Function with Examples”

325.7 FDRMOVE CONSOLIDATE VOLUMES WITH MOVE FUNCTION

All examples in this section are found in the JCL library installed with FDRMOVE. The member names are PA32507*x*.

CONSOLIDATE ENTIRE VOLUMES WITHOUT INSTANT REPLICATION (MOVE) EXAMPLE

In this scenario, you want to consolidate volumes with FDRMOVE, but you do not have any instant replication facility available. You can also use it when you know that the data sets involved are not always in use, so that FDRMOVE has an opportunity to move them.

Step 1: (Simulation) of the Consolidate Entire Volumes without Instant Replication (MOVE) Example

A simulated MOVE that identifies all volumes and data sets to be moved. It also identifies the jobs currently holding the SYSDSN enqueue on those data sets. It also identifies catalogs and other data sets that FDRMOVE cannot move; these volumes may need to be moved with an FDRPAS swap.

```
//jobname  JOB …
//STEP1    EXEC PGM=FDRMOVE,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//FDRSUMM  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  SIMMOVE  TYPE=DSF,DISABLENEW=YES
  SELECT   ALLDSN,VOL=IN0001,NVOL=(OUT001,OUT002,…,OUT022)
  SELECT   ALLDSN,VOL=IN0002,NVOL=(OUT001,OUT002,…,OUT022)
  SELECT   ALLDSN,VOL=IN0003,NVOL=(OUT001,OUT002,…,OUT022)
  …
  SELECT   ALLDSN,VOL=IN0060,NVOL=(OUT001,OUT002,…,OUT022)
/*
```

Repeat Step 1 if necessary to fix JCL or control statement errors, NVOL= lists, and the rest.

Step 2: (MOVE Process) of the Consolidate Entire Volumes without Instant Replication (MOVE) Example

The MOVE job. It moves the data sets from the input volumes when they become available (not enqueued). DISABLENEW=YES sets the input volumes to DISNEW (SMS) or PRIVATE (non SMS-managed) so that the volumes are no longer used for new data sets; remove it if you do not plan to remove the input subsystem.

```
//jobname  JOB …
//STEP2    EXEC PGM=FDRMOVE,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//FDRSUMM  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  MOVE     TYPE=DSF,DISABLENEW=YES
  SELECT   ALLDSN,VOL=IN0001,NVOL=(OUT001,OUT002,…,OUT022)
  SELECT   ALLDSN,VOL=IN0002,NVOL=(OUT001,OUT002,…,OUT022)
  SELECT   ALLDSN,VOL=IN0003,NVOL=(OUT001,OUT002,…,OUT022)
  …
  SELECT   ALLDSN,VOL=IN0060,NVOL=(OUT001,OUT002,…,OUT022)
/*
```

NOTE: You can leave the MOVE job running for an indefinite period, until all data sets have been moved. You can use the console STOP (P) command to terminate it and restart it later, if necessary. The STOPAFTER= operand automatically stops the MOVE after the specified period.

NOTE: For maximum performance, you should submit multiple MOVE jobs, with unique job names, each specifying a subset of your input volumes. This allows FDRMOVE to process eight input volumes concurrently per job that allows the moves to complete much faster.

Step 3: (Check Data Sets) of the Consolidate Entire Volumes without Instant Replication (MOVE) Example

Use the console STATUS command or the ISPF panel to show what jobs are holding the remaining data sets.

Step 4: (Restart Applications) of the Consolidate Entire Volumes without Instant Replication (MOVE) Example

"Bounce" the applications that are using the remaining data sets, if necessary to get them moved.

Step 5: (Verify Data Sets Copied) of the Consolidate Entire Volumes without Instant Replication (MOVE) Example

(Optional). If your intention is to empty the input volumes so that the source subsystem can be shutdown, use SIMMOVE with the `VTOCEMPTY=CHECK` operand and `SELECT ALLDSN` to verify that all data sets have been moved from the indicated volumes.

```
//jobname  JOB …
//STEP5    EXEC PGM=FDRMOVE,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//FDRSUMM  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  SIMMOVE  TYPE=DSF,VTOCEMPTY=CHECK
  SELECT   ALLDSN,VOL=IN0001
  SELECT   ALLDSN,VOL=IN0002
  SELECT   ALLDSN,VOL=IN0003
  …
  SELECT   ALLDSN,VOL=IN0060
/*
```

Step 6: (FDRPAS) of the Consolidate Entire Volumes without Instant Replication (MOVE) Example

(Optional). Run an FDRPAS non-disruptive swap if there are data sets remaining that cannot be freed.

325.8 FDRMOVE MOVE ACTIVE DATA SETS WITH FASTMOVE

All examples in this section are found in the JCL library installed with FDRMOVE. The member names are PA32508*x*.

MOVE ACTIVE DATA SET WITH INSTANT REPLICATION (FASTMOVE) EXAMPLE

In this scenario, you want to move the data sets used by two long-running applications, each using a separate SMS storage group. `STORGRP=` is used to select the volumes where these applications reside. Offline DASD volumes in the range 7F00-7FFF are used as transit stations.

NOTE: The scenario below can be modified to replace STORGRP= with:
`SELECT ALLDSN,VOL=` – to select all data sets in the specified volume(s).
`SELECT CATDSN=`*mask* – to select specified data sets and volumes from the catalog.

Step 0: (Add Output Volumes to Input Storage Groups) of the Move Active Data Set with Instant Replication (FASTMOVE) Example

Is to add the output volumes to each of the input SMS storage groups. There must be sufficient space in the new volumes to accommodate all data sets from the old volumes.

Step 1: (MOVE Process) of the Move Active Data Set with Instant Replication (FASTMOVE) Example

Is a MOVE job similar to this. This example moves inactive data sets (and those that become inactive) but only if they are 100 cylinders in size or less. Data sets are moved from the volumes in the two storage groups in the old control unit (`CUID=17642`) to the storage group in the new control unit (`NEWCUID=57232`). Note that if you are sure that all of the selected data sets will be active until the applications are bounced, you might want to skip this step.

```
//jobname  JOB
//STEP1    EXEC PGM=FDRMOVE,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//FDRSUMM  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  MOVE     TYPE=DSF
  EXCLUDE  ALLDSN,SIZE=1500  EXCLUDE DATA SETS > 100 CYLINDERS
  SELECT   ALLDSN,CUID=17642,STORGRP=CICSGRP,
           NEWCUID=57232,NEWSTORGRP=CICSGRP
  SELECT   ALLDSN,CUID=17642,STORGRP=PAYGRP,
           NEWCUID=57232,NEWSTORGRP=PAYGRP
/*
```

You can submit Step 1 some time before you intend to bounce the applications, perhaps days before. You can terminate it when you are ready to run the FASTMOVE with a console `STOP (P)` command. You can also specify `STOPAFTER=` to automatically terminate after the specified time.

Step 2: (SIMMOVE) of the Move Active Data Set with Instant Replication (FASTMOVE) Example

Is a simulated MOVE that identifies all remaining volumes and data sets to be moved. It also identifies the jobs currently holding the SYSDSN enqueue on those data sets, shows the minimum number of NVOLs required, and it shows the number of transit stations required. It also validates the internal FDRPAS TRANSIT job.

```
//jobname  JOB
//STEP2    EXEC PGM=FDRMOVE,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//FDRSUMM  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  SIMMOVE  TYPE=DSF,DISABLENEW=YES
  SELECT   ALLDSN,CUID=17642,STORGRP=CICSGRP,
           NEWCUID=57232,NEWSTORGRP=CICSGRP
  SELECT   ALLDSN,CUID=17642,STORGRP=PAYGRP,
           NEWCUID=57232,NEWSTORGRP=PAYGRP
/*
//PASJOB   DD DATA,DLM=$$
//TRANSIT1 JOB (accounting info),FDRMOVE,CLASS=c,
//         MSGCLASS=m  <-- add any other required JOB parameters
//PASTRANS EXEC PGM=FDRPAS,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//FDRSUMM  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  SWAP     TYPE=TRANSIT
  MOUNT    VOL=&&&&&&,SWAPUNIT=7F*
$$
```

Repeat Step 2 if necessary to fix JCL or control statement errors, NVOL= and SWAPUNIT= lists, and the rest.

Step 3: (FDRPAS Monitor) of the Move Active Data Set with Instant Replication (FASTMOVE) Example

Is an FDRPAS MONITOR task that must execute on all LPARs (except the LPAR where the SWAP TYPE=TRANSIT job will run), specifying the offline transit station DASD volumes. You may need a unique job name for each MONITOR. For example,

```
//jobname  JOB
//STEP3    EXEC PGM=FDRPAS,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  MONITOR  TYPE=SWAP
  MOUNT    SWAPUNIT=7F*
/*
```

See Section 310.23 "MONITOR SWAP Examples" for other ways to start the MONITOR tasks.

Step 4: (FASTMOVE Job) of the Move Active Data Set with Instant Replication (FASTMOVE) Example

Is the FASTMOVE job, same as Step 2 except changing SIMMOVE to FASTMOVE. Submit it sometime before you intend to bounce the application, to give FDRPAS time to move all the volumes into the transit stations. DISABLENEW=YES sets the input volumes to DISNEW (SMS) so that the volumes are no longer used for new data sets. If you plan to keep the input subsystem, you can re-enable the input volumes after the move.

```
//jobname  JOB
//STEP4    EXEC PGM=FDRMOVE,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//FDRSUMM  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  FASTMOVE TYPE=DSF,DISABLENEW=YES
  SELECT   ALLDSN,CUID=17642,STORGRP=CICSGRP,
           NEWCUID=57232,NEWSTORGRP=CICSGRP
  SELECT   ALLDSN,CUID=17642,STORGRP=PAYGRP,
           NEWCUID=57232,NEWSTORGRP=PAYGRP
/*
//PASJOB   DD DATA,DLM=$$
//TRANSIT1 JOB (accounting info),FDRMOVE,CLASS=c,
//         MSGCLASS=m  <-- add any other required JOB parameters
//PASTRANS EXEC PGM=FDRPAS,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//FDRSUMM  DD SYSOUT=*
//SYSIN    DD *
  SWAP     TYPE=TRANSIT
  MOUNT    VOL=&&&&&&,SWAPUNIT=7F*
$$
```

The internal TRANSIT1 job is submitted by FASTMOVE after it identifies all volumes that must be moved to a transit station (if any).

As each volume reaches a transit station, FASTMOVE begins testing the selected data sets on that volume to see if they are active (enqueued). Data sets that are inactive or become inactive are quickly moved with instant replication technology.

Step 5: (Check Status) of the Move Active Data Set with Instant Replication (FASTMOVE) Example

Is to use the console STATUS command or the ISPF panel to monitor when the volumes are all in transit stations. In the sample display below, no volumes are awaiting transit or moving to transit and 250 volumes are in transit stations.

```
      DATASETS:              --------NUMBER OF VOLUMES-----------
    MOVED  ACTIVE   AWAITING TRANSIT MOVING TO TRANSIT IN TRANSIT STATION
--------- -------   ---------------- ----------------- ------------------
     5282     432            0                0                 250
```

The STATUS display also shows what jobs are holding the data sets. In this example it shows that data sets are being held by a CICS region.

```
ACTIVE DATASETS FOR VOLUME SH20E0     NUMBER JOBS  JOBNAMES HOLDING ENQUEUE
------------------------------------ -----------  --------------------------
DB2.PAYROLL.VSH20E0.VSAM.CLUSTER              1   (CICS101)
DB2.DSNDBC.DSNDB06.DSNKKX02.I0001.A001        1   (CICS101)
```

Step 6: ("Bounce" Applications) of the Move Active Data Set with Instant Replication (FASTMOVE) Example

"Bounce" the applications (together or at separate times) when the status shows that all volumes are in transit stations.

Step 7: (Restart Applications) of the Move Active Data Set with Instant Replication (FASTMOVE) Example

Restart the applications. If some of the data sets were not moved, you can leave FASTMOVE running and bounce the application again later. Alternatively, you can `STOP (P)` the FASTMOVE job and restart it at a later time if you can.

Step 8: (Check Volumes Moved) of the Move Active Data Set with Instant Replication (FASTMOVE) Example

(Optional.) If your intention is to empty the input volumes so that the source subsystem can be shutdown use SIMMOVE with the `VTOCEMPTY=CHECK` operand and `SELECT ALLDSN` to verify that all data sets have been moved from the indicated volumes.

```
//jobname  JOB
//STEP8    EXEC PGM=FDRMOVE,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//FDRSUMM  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  SIMMOVE  TYPE=DSF,VTOCEMPTY=CHECK
  SELECT   ALLDSN,CUID=17642,STORGRP=CICSGRP
  SELECT   ALLDSN,CUID=17642,STORGRP=PAYGRP
/*
```

Step 9: (Check Data Sets Moved) of the Move Active Data Set with Instant Replication (FASTMOVE) Example

(Optional.) If you wish to ensure that all selected data sets were moved, use SIMMOVE with the `VTOCEMPTY=CHECK` operand and `SELECT CATDSN=` to verify that the selected data sets have been moved to the target volumes.

```
//jobname  JOB
//STEP9    EXEC PGM=FDRMOVE,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//FDRSUMM  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  SIMMOVE  TYPE=DSF,VTOCEMPTY=CHECK
  SELECT   CATDSN=PAY.APPL1.**,NEWCUID=57232,NEWSTORGRP=PAYGRP
  SELECT   CATDSN=CICS.APPL2.**,NEWCUID=57232,NEWSTORGRP=CICSGRP
/*
```

325.9 CONSOLIDATE VOLUMES WITH FDRPAS AND FASTMOVE

All examples in this section are found in the JCL library installed with FDRMOVE. The member names are PA32509*x*.

CONSOLIDATE ENTIRE VOLUMES WITH FDRPAS AND INSTANT REPLICATION EXAMPLE

In this scenario, you want to consolidate entire volumes to a smaller set of larger volumes, such as 3390-9 to 3390-27. FDRPAS is used to move 1/3 of the volumes to the new devices non-disruptively. This assumes that the VTOCs on the source volumes are already large enough to hold three times the data sets after the consolidation. The target volumes are in the range 7000-70FF and are currently offline.

If you are licensed for FDRABR or FDREPORT (or have a trial copy of FDRMOVE), you can use this FDREPORT step to report on the VTOC sizes and VTOC free space on the volumes you want to move with FDRPAS. If the VTOCs are large enough already, proceed to Step1 below. If not, you may be able to expand the VTOCs with the EXPANDVTOC function of FDRPAS, before or after you SWAP the volume.

```
//VTOCREPT EXEC PGM=FDREPORT,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdr.loadlib
//SYSPRINT DD SYSOUT=*
//ABRMAP   DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  XSELECT  VOL=IN0*
  REPORT   FIELD=(VLVOLSER,VLUNIT,VLDEVTYP,VLDSCB,VL%FDSCB,
                  VLINDSTA,VLTOTVIR,VL%FINDX)
  PRINT    DATATYPE=VOLDATA
/*
```

Step 0: (EXPANDVTOC) of the Consolidate Entire Volumes with FDRPAS and Instant Replication Example

(Optional.) This step expands the VTOCs to 650 tracks on volumes that are swapped with FDRPAS and need larger VTOCs. You also need to submit FDRPAS MONITOR tasks for all other LPARs as documented under EXPANDVTOC. Only one offline `SWAPUNIT=` is needed for all volumes.

```
//STEP0    EXEC PGM=FDRPAS,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  EXPANDVTOC  TYPE=FULL
  MOUNT    VOL=IN0001,SWAPUNIT=07C3,VTOCSIZE=650
  MOUNT    VOL=IN0002,SWAPUNIT=07C3,VTOCSIZE=650
  …
  MOUNT    VOL=IN0022,SWAPUNIT=07C3,VTOCSIZE=650
/*
```

After the FDRPAS swaps, FASTMOVE is used to copy data sets from the remaining 2/3 of the input volumes to complete the consolidation. Offline DASD volumes in the range 7F00-7FFF are used as transit stations.

Step 1: (FDRPAS Monitor) of the Consolidate Entire Volumes with FDRPAS and Instant Replication Example

This step is the FDRPAS MONITOR task that must execute on all LPARs (it does not need to run on the LPAR where the SWAP TYPE=TRANSIT runs). It specifies the offline target DASD volumes for the FDRPAS swaps and also the offline transit station DASD volumes for FASTMOVE. The same MONITOR task can handle both types of DASD volumes so leave it running until the FASTMOVE jobs are complete. You may need a unique job name for each MONITOR task. For example,

```
//STEP1    EXEC PGM=FDRPAS,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//FDRSUMM  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  MONITOR  TYPE=SWAP
  MOUNT    SWAPUNIT=70*  <-- targets for FDRPAS
  MOUNT    SWAPUNIT=7F*  <-- targets for FDRPAS TRANSIT
/*
```

See Section 310.23 "MONITOR SWAP Examples" for other ways to start the MONITOR tasks.

Step 2: (FDRPAS SWAP) of the Consolidate Entire Volumes with FDRPAS and Instant Replication Example

This step is the FDRPAS SWAP task to non-disruptively move 1/3 of the input DASD volumes; choose the volumes with the most active data sets. If any volumes contain data sets that cannot be moved by FDRMOVE, such as catalogs, you should select those volumes. Two extra volumes are moved to handle special conditions with multi-volume data sets and very full volumes.

```
//STEP2    EXEC PGM=FDRPAS,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//FDRSUMM  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  SWAP     TYPE=FULL
  MOUNT    VOL=IN0001,SWAPUNIT=7001
  MOUNT    VOL=IN0002,SWAPUNIT=7002
  MOUNT    VOL=IN0003,SWAPUNIT=7003
  …
  MOUNT    VOL=IN0021,SWAPUNIT=7021
  MOUNT    VOL=IN0022,SWAPUNIT=7022
/*
```

Step 3: (SIMMOVE) of the Consolidate Entire Volumes with FDRPAS and Instant Replication Example

This step is a simulated MOVE that identifies data sets to be moved on the remaining input volumes. It also identifies any jobs currently holding the SYSDSN enqueue on those data sets, shows the minimum number of NVOLs required, and it shows the number of transit stations required. It also validates the internal FDRPAS TRANSIT job.

```
//STEP3    EXEC PGM=FDRMOVE,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//FDRSUMM  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  SIMMOVE TYPE=DSF,DISABLENEW=YES
  SELECT  ALLDSN,VOL=IN0023,NVOL=(IN0001,IN0002,…,IN0022)
  SELECT  ALLDSN,VOL=IN0024,NVOL=(IN0001,IN0002,…,IN0022)
  SELECT  ALLDSN,VOL=IN0025,NVOL=(IN0001,IN0002,…,IN0022)
  …
  SELECT  ALLDSN,VOL=IN0059,NVOL=(IN0001,IN0002,…,IN0022)
  SELECT  ALLDSN,VOL=IN0060,NVOL=(IN0001,IN0002,…,IN0022)
/*
//PASJOB   DD DATA,DLM=$$
//TRANSIT1 JOB (accounting info),FDRMOVE,CLASS=c,
//         MSGCLASS=m  <-- add any other required JOB parameters
//PASTRANS EXEC PGM=FDRPAS,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//FDRSUMM  DD SYSOUT=*
//SYSIN    DD *
  SWAP     TYPE=TRANSIT
  MOUNT    VOL=&&&&&&,SWAPUNIT=7F*
$$
```

Repeat Step 3 if necessary to fix JCL or control statement errors, `NVOL=` and `SWAPUNIT=` lists, and the rest.

Step 4: (FASTMOVE) of the Consolidate Entire Volumes with FDRPAS and Instant Replication Example

This step is the FASTMOVE job. Using the Step 3 job, except changing SIMMOVE to FASTMOVE. Submit it sometime before you intend to bounce the application, to give FDRPAS time to move all the volumes into the transit stations. Note that the NVOL= list specifies the first ⅓ of the input volumes that were moved by FDRPAS.

```
//STEP4    EXEC PGM=FDRMOVE,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//FDRSUMM  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  FASTMOVE TYPE=DSF,DISABLENEW=YES
  SELECT  ALLDSN,VOL=IN0023,NVOL=(IN0001,IN0002,…,IN0022)
  SELECT  ALLDSN,VOL=IN0024,NVOL=(IN0001,IN0002,…,IN0022)
  SELECT  ALLDSN,VOL=IN0025,NVOL=(IN0001,IN0002,…,IN0022)
  …
  SELECT  ALLDSN,VOL=IN0059,NVOL=(IN0001,IN0002,…,IN0022)
  SELECT  ALLDSN,VOL=IN0060,NVOL=(IN0001,IN0002,…,IN0022)
/*
//PASJOB   DD DATA,DLM=$$
//TRANSIT1 JOB (accounting info),FDRMOVE,CLASS=c,
//         MSGCLASS=m  <-- add any other required JOB parameters
//PASTRANS EXEC PGM=FDRPAS,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//FDRSUMM  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  SWAP     TYPE=TRANSIT
  MOUNT    VOL=&&&&&&,SWAPUNIT=7F*
$$
```

The internal TRANSIT1 job is submitted by FASTMOVE after it identifies all volumes that must be moved to a transit station (if any).

As each volume reaches a transit station, FASTMOVE begins testing the selected data sets on that volume to see if they are active (enqueued). Data sets that are inactive or become inactive are quickly moved with instant replication technology.

Step 5: (Check STATUS) of the Consolidate Entire Volumes with FDRPAS and Instant Replication Example

Use the console STATUS command or the ISPF panel to monitor when the volumes are all in transit stations. In the sample display below, no volumes are awaiting transit or moving to transit and 250 volumes are in transit stations.

```
      DATASETS:              --------NUMBER OF VOLUMES-----------
   MOVED  ACTIVE   AWAITING TRANSIT MOVING TO TRANSIT IN TRANSIT STATION
-------- -------   ---------------- ----------------- ------------------
    5282     432            0                0                250
```

The STATUS display also shows what jobs are holding any active data sets. In this example, it shows that data sets are being held by a batch job and a TSO user.

```
ACTIVE DATASETS FOR VOLUME SH20E0     NUMBER JOBS  JOBNAMES HOLDING ENQUEUE
------------------------------------ -----------  --------------------------
DB2.PAYROLL.VSH20E0.VSAM.CLUSTER              1   (PAYCHKS)
USER32.PROFILE                                1   (USER32)
```

Step 6: ("Bounce" Applications) of the Consolidate Entire Volumes with FDRPAS and Instant Replication Example

"Bounce" applications that are holding data sets to be moved when the status shows that all volumes are in transit stations. Alternately, you can simply wait for the data sets to be freed if they are likely to be freed by batch job completions and TSO logoffs.

Step 7: (Restart Applications) of the Consolidate Entire Volumes with FDRPAS and Instant Replication Example

Restart any applications you bounced. If some of the data sets were not moved, you can leave FASTMOVE running and bounce the application again later. Alternatively, you can `STOP (P)` the FASTMOVE job and restart it at a later time if you can.

Step 8: (Verify Moved Data Sets) of the Consolidate Entire Volumes with FDRPAS and Instant Replication Example

(Optional.) If your intention is to empty the input volumes so that the source subsystem can be shutdown, use SIMMOVE with the `VTOCEMPTY=CHECK` operand and `SELECT ALLDSN` to verify that all data sets have been moved from the indicated volumes. Do this check only on the volumes that were not moved by FDRPAS in Step 2.

```
//STEP8    EXEC PGM=FDRMOVE,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSPRINT DD SYSOUT=*
//FDRSUMM  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  SIMMOVE  TYPE=DSF,VTOCEMPTY=CHECK
  SELECT   ALLDSN,VOL=IN0023
  SELECT   ALLDSN,VOL=IN0024
  SELECT   ALLDSN,VOL=IN0025
  …
  SELECT   ALLDSN,VOL=IN0059
  SELECT   ALLDSN,VOL=IN0060
/*
```

325.10 FDRMOVE CONSOLE COMMANDS AND ISPF INTERFACE

FDRMOVE CONSOLE COMMANDS

An FDRMOVE job accepts `STOP (P)` and `MODIFY (F)` commands from the console.

STOP AND CANCEL COMMANDS

To stop FDRMOVE, issue the `STOP (P)` command with the name of the FDRMOVE job

```
P jobname
```

FDRMOVE finishes moving all data sets that are actively being moved, and then terminates cleanly.

You can also terminate an FDRMOVE job with a console `CANCEL (C)` command. However, the first `CANCEL (C)` you issue is intercepted and acts as if a `STOP (P)` command was entered. Message FDR267 is issued to tell you that you can cause immediately cancellation by entering a second `CANCEL (C)` command.

RECOMMENDATION: ***INNOVATION DATA PROCESSING strongly recommends against canceling an FDRMOVE job. Data sets that were being moved at the time of the CANCEL (C) may be left in an unknown state. It is better to STOP (P) the job and let it finish gracefully.***

If a FASTMOVE job has submitted the FDRPAS job to move volumes to the transit stations, stopping the FASTMOVE job also stops the FDRPAS job, but only after all volumes have been returned to their original devices (see `TRANSITRETURN=NO` below).

If you submitted the FDRPAS transit station job independently or you had to cancel the FDRMOVE job, you can also issue a `STOP (P)` command to the FDRPAS job.

MODIFY STATUS COMMAND

You can request the status of an FDRMOVE job by entering:

```
F jobname,STATUS    (or just STA)
```

For a MOVE job, this displays the volumes that still have data sets to be processed, the active (enqueued) data sets for which the step is waiting and a summary of data sets processed. Here is a sample of the STATUS output for MOVE:

```
FDR265  MODIFY COMMAND COMPLETED - STATUS
FDR184  VOL=SH20E0 IN TRANSIT STATION EMC00547 -- EMC00547       2 DATASETS TO MOVE - HIGHEST SEQ#   1
FDR184          |---FOR ACTIVE DSN=TEST1.PDS.CNTL                                  1 JOBS(TSO123)
FDR184          |---FOR ACTIVE DSN=TSO123.PROFILE                                  1 JOBS(TSO123)
FDR184  FDR MOVED         24 DATASETS/COMPONENTS SUCCESSFULLY
FDR184  MOVE STATUS        2 ACTIVE DATASETS    0 AWAITING TRANSIT    0 MOVING TO TRANSIT    0 IN TRANSIT STATION
```

For a FASTMOVE job, this displays the status of each volume still containing data sets to be processed (in a transit station or not), active (enqueued) data sets and a summary of data sets processed. Here is a sample of the STATUS output for FASTMOVE:

```
FDR265  MODIFY COMMAND COMPLETED - STATUS
FDR184  VOL=SH20E0 IN TRANSIT STATION EMC00547 -- EMC00547       4 DATASETS TO MOVE - HIGHEST SEQ#   1
FDR184          |---FOR ACTIVE DSN=ICF1.MOVE013.VSH20E0.KSDS.A0002.CLUSTER         1 JOBS(MOVE013E)
FDR184          |---FOR ACTIVE DSN=ICF1.MOVE013.VSH20E0.PO.A0002                   1 JOBS(MOVE013E)
FDR184          |---FOR ACTIVE DSN=ICF1.MOVE013.VSH20E0.PS.A0001                   2 JOBS(MOVE013Q,MOVE013E)
FDR184          |---FOR ACTIVE DSN=ICF1.MOVE013.VSH20E0.PS.A0002                   1 JOBS(MOVE013E)
FDR184  VOL=SH20C0  MOVING TO TRANSIT IBM76421 TO EMC00547      28 DATASETS TO MOVE - HIGHEST SEQ#   1
FDR184  VOL=SH20C3 IN TRANSIT STATION EMC00547 -- EMC00547       1 DATASETS TO MOVE - HIGHEST SEQ#   1
FDR184          |---FOR ACTIVE DSN=ICF1.MOVE013.VSH20C3.PS.A0001                   1 JOBS(MOVE013Q)
FDR184  FDR MOVED          0 DATASETS/COMPONENTS SUCCESSFULLY
FDR184  MOVE STATUS       33 ACTIVE DATASETS    0 AWAITING TRANSIT    1 MOVING TO TRANSIT    2 IN TRANSIT STATION
```

On the volume line, the source and target DASD manufacturer and serial number are shown (for example, IBM76421). If a volume is in a transit station, the source and target IDs are the same. The "HIGHEST SEQ#" shows the status of multi-volume data sets on the volume; a value of 1 indicates no multi-volume data sets were found, but a value greater than 1 indicates the highest sequence number found on this volume. If you locate the highest sequence number displayed for any volume, this indicates the minimum number of NVOLs you need to move these multi-volume data sets. The sequence number may be followed by "NOIX" indicating that there is no active VTOCIX on the volume; this impacts FDRMOVE performance when it is scratching data sets from the volume, so you may wish to create a VTOCIX.

For both types of STATUS displays, the active data set display shows how many jobs currently hold the SYSDSN enqueue on the data set, and the first few job names are shown.

The display may also show "* CHECK ERROR MESSAGES *" if any error messages have occurred, or "* MOVE IS SUSPENDED *" if the job has been suspended.

STATUS displays a maximum of 100 lines in your JOBLOG (these may be echoed on your console). The STATUS is also put to SYSPRINT, displaying up to 1000 lines.

If you want to limit the volume status without details on the data sets, enter:

```
F jobname,STATUS NODSN
```

If you want to limit the volume status to certain volumes, enter:

```
F jobname,STATUS VOL=volser
```

"*volser*" can be a specific volume serial number, a volume serial number prefix followed by an asterisk (ABC*), or just an asterisk to display all volumes (the default).

You can specify both parameters, in either order, for example,

```
F jobname,STATUS VOL=volser,NODSN
```

CONSOLE MODIFY SUSPEND AND RESUME

You can issue console commands:

```
F jobname,SUSPEND
```

or

```
F jobname,RESUME
```

To any active FDRMOVE or FDRPAS TRANSIT job. SUSPEND completes any active operations before going into a suspend state. RESUME starts the job going again.

SUSPEND to an FDRMOVE job finishes any existing MOVE task and does not start any new move task until a MODIFY RESUME. This allows the user to have an FDRMOVE job running for a long time, but have it moving data sets only during certain periods of time. For instance the user does the SUSPEND at 8AM and resumes it 12AM the next morning so that the moves are only taking place between 12AM and 8AM overnight. The SUSPEND can also be done from the FDRMOVE ISPF panels.

If the SUSPEND/RESUME is issued to the FDRPAS TRANSIT job, all active swaps to transit complete and no new ones start until the RESUME is done. This is different than the SUSPEND/RESUME command on the FDRPAS ISPF panels that temporary stop any tracks from being copied on active swap volumes.

MODIFY TRANSITRETURN COMMAND

To dynamically modify the value of TRANSITRETURN=, issue the MODIFY (F) command, specifying the job name of an FDRPAS "SWAP TYPE=TRANSIT" job:

```
F jobname,TRANSITRETURN=NO
```

By default, FASTMOVE returns the source volumes to their original DASD devices after FASTMOVE is done with them. This option causes FASTMOVE to leave the source volumes in the transit station. It is the same as specifying TRANSITRETURN=NO on the SWAP statement.

If you terminate a FASTMOVE job (see the STOP (P) command described in "STOP and CANCEL Commands" in Section 325.10), the FASTMOVE job automatically terminates the associated FDRPAS job, but only after FDRPAS has moved all volumes back to their original devices. If the FASTMOVE job has not completed all data sets and you intend to restart it later, you should use this procedure:

1. Enter a console command: F *pasjob*,TRANSITRETURN=NO
2. Enter a console command: P *movejob*

This terminates the FASTMOVE job (after it finishes any data sets actively being moved) and terminates the FDRPAS job, leaving any volumes that are in the transit stations in those stations. Later when the FASTMOVE job is restarted, FDRMOVE recognizes that the volumes are already in the same control unit as the targets, and does not move them again. As long it is run on the same LPAR and an intervening IPL has not occurred, it returns the volumes to their original devices.

FDRMOVE ISPF INTERFACE

The FDRMOVE ISPF interface allows you to monitor FDRMOVE operations on the system where your TSO session is logged on.

INVOKING THE FDRMOVE ISPF DIALOGS

The FDRMOVE ISPF dialogs are integrated with the ISPF dialogs of FDR, the INNOVATION DATA PROCESSING DASD management system. If you have installed the FDR ISPF dialogs at a release level matching the level of FDRMOVE that you are using, your FDR or ABR main ISPF menu has an option "M" for FDRMOVE. If so, you can skip the invocation of ABRALLOC shown below and use that option to invoke the FDRMOVE dialogs.

If you do not have an appropriate level of the FDR dialogs installed, or they are at a lower release level than the level of FDRMOVE in use, then use the following procedure:

Go to ISPF option 6, or exit ISPF to TSO READY mode, and issue this command:

```
EXEC 'fdrpas.clist.library(ABRALLOC)'
```

Specifying the name of the FDRPAS CLIST library that was installed during the installation of FDRPAS and FDRMOVE (see Section 380.5 "Invoking the Installation ISPF Dialog (Panel A.I)"). This allocates all of the required FDRPAS ISPF libraries and invokes the ABR dialogs (including FDRMOVE).

If you issued the ABRALLOC from TSO READY, you see a standard ISPF main menu with an additional option "A" (for ABR). If your installation has modified your ISPF main menu, this may look considerably different than you are normally used to seeing. This ISPF menu looks something like this:

ISPF PRIMARY OPTION MENU

```
  Menu  Utilities  Compilers  Options  Status  Help
------------------------------------------------------------------------------
                           ISPF Primary Option Menu
Option ===> A

0  Settings      Terminal and user parameters          User ID . : USER1
1  View          Display source data or listings       Time. . . : 09:05
2  Edit          Create or change source data          Terminal. : 3278
3  Utilities     Perform utility functions             Screen. . : 1
4  Foreground    Interactive language processing       Language. : ENGLISH
5  Batch         Submit job for language processing    Appl ID . : ISR
6  Command       Enter TSO or Workstation commands     TSO logon : VnnISPF
7  Dialog Test   Perform dialog testing                TSO prefix: USER1
8  LM Facility   Library administrator functions       System ID : CPUB

9  IBM Products  IBM program development products      MVS acct. : **NONE**
10 SCLM          SW Configuration Library Manager      Release . : ISPF n.n
11 Workplace     ISPF Object/Action Workplace
A  FDR/ABR       FDR/ABR DASD Management Functions
```

Select option "A" on the Option line, as shown and press ENTER to display the FDR/ABR primary options menu.

FDR PRIMARY OPTIONS MENU

The FDR Primary Options Menu looks similar to this:

FDR PRIMARY OPTIONS MENU – PANEL A

```
---------- FDR TOTAL DASD MANAGEMENT SYSTEM -- FDR PRIMARY OPTIONS MENU -------
 OPTION  ===> M

  1  REPORTS     - ABR REPORTING FUNCTIONS
  2  RESTORE     - ABR DATA SET RESTORE
  3  ARCHIVE     - ABR DATA SET ARCHIVE OR SUPERSCRATCH
  4  BACKUP      - ABR DATA SET BACKUP
  5  REMOTE Q    - ABR REMOTE QUEUE UTILITY FUNCTIONS

  C  COMPAKTOR   - COMPAKTOR MAP AND SIMULATION REPORTS
  R  RELEASE     - COMPAKTOR RELEASE

  I  INSTALL     - INSTALLATION AND MAINTENANCE OF FDR AND OPTIONAL PRODUCTS
  J  JCL PARMS   - SPECIFY FDR JCL AND SYSOUT DEFAULTS FOR SUBMITTED JOBS
  K  FORMAT      - MODIFY FORMAT OF GENERATED REPORTS

  P  PLUG & SWAP - FDRPAS PLUG & SWAP
  E  FDRERASE    - FDR DISK ERASE
  M  FDRMOVE     - FDRMOVE PLUG & SWAP DATA SET MOVE

  MC MESSAGES    - FDR MESSAGES AND CODES QUERY FACILITY
  Q  QUERY       - FDR/ABR STATISTICS QUERY
  S  SRS         - SEARCH, REPORT, SERVICES DIALOG
  T  FDRTSEL     - BACKUP FILE MANAGEMENT UTILITY
```

Select option “M” to monitor FDRMOVE. Other options are used for FDRABR and other components of the FDR DASD management software. Option “I” is used during installation of FDRPAS as described in Section 380.5 “Invoking the Installation ISPF Dialog (Panel A.I)”.

FDRMOVE PANEL

This panel is displayed by the "M" option. It is used to monitor FDRMOVE jobs, using MOVE or FASTMOVE, with information similar to the console `F movejob,STATUS` described earlier. The FDRMOVE panel has these input fields:

Display DSN – Is set to YES or NO and controls whether the display contains lists of data sets that FDRMOVE is waiting to move. YES is the default

Jobname – Can be used to limit the display to certain FDRMOVE jobs if more than one is running. You can specify a specific FDRMOVE job name, a job name prefix followed by an asterisk (MOVE*), or just an asterisk to display all FDRMOVE jobs (the default).

Vol – Can be used to limit the display to certain volumes that FDRMOVE is processing. You can specify a specific volume serial number, a volume serial number prefix followed by an asterisk (ABC*), or just an asterisk to display all volumes (the default).

Refresh – Enter a count (0 to 99) and the FDRMOVE display is refreshed every few seconds until the count goes to 0 again.

The "Display DSN" and "Jobname" fields are remembered in your ISPF profile for the next time you invoke the panel.

The ISPF panel accepts a single command on the COMMAND line. You may enter "*command jobname*" where "*command*" is either "STOP", "SUSPEND", or "RESUME" and "*jobname*" is a specific FDRMOVE job name or a prefix (for example, "STOP MOVE*"). The STOP, SUSPEND, and RESUME commands verify that the selected job names are FDRMOVE jobs and internally issue the appropriate STOP, SUSPEND, or RESUME command for each one.

FDRMOVE PLUG & SWAP DATA SET MOVE – PANEL A.M

```
---------------------- FDRMOVE PLUG & SWAP DATA SET MOVE ---------- Row 1 of 11
COMMAND ===>                                                  SCROLL ===> PAGE

Display DSN ===> YES  Jobname ===> *          Vol ===> *      Refresh ===> 0

-- MOVE STATUS REPORT FOR JOB MOVE013  AT 15:00:32 --

      DATASETS:              --------NUMBER OF VOLUMES-----------
    MOVED  ACTIVE   AWAITING TRANSIT MOVING TO TRANSIT IN TRANSIT STATION
--------- -------   ---------------- ----------------- ------------------
        0      84           1                2                  0

VOL=SH20C0   AWAITING TRANSIT IBM76421 TO EMC00547      28 DATASETS TO MOVE
VOL=SH20E0  MOVING TO TRANSIT IBM76421 TO EMC00547      55 DATASETS TO MOVE
VOL=SH20C3  MOVING TO TRANSIT IBM76421 TO EMC00547       1 DATASETS TO MOVE
```

In the example above, the FASTMOVE job has 84 data sets to move from three DASD volumes. Data set names are not yet displayed because the volumes are not yet in the transit stations.

The FASTMOVE job has requested that the FDRPAS transit station job move the three volumes. Two volumes are being actively moved to their transit stations, and one is waiting move. Note that the display shows the vendor and serial number of the input and output DASD subsystems.

As soon as each volume gets to its transit station, FASTMOVE begins testing the status of the SYSDSN enqueue on each selected data set. As soon as the data set is no longer

active, FASTMOVE moves that data set (using “instant replication” technology).

FDRMOVE PLUG & SWAP DATA SET MOVE – PANEL A.M

```
-------------------- FDRMOVE PLUG & SWAP DATA SET MOVE ---------- Row 1 of 19
COMMAND ===>                                                  SCROLL ===> PAGE

Display DSN ===> YES  Jobname ===> *                         Refresh ===> 0

-- MOVE STATUS REPORT FOR JOB MOVE013  AT 15:01:22 --

      DATASETS:              --------NUMBER OF VOLUMES-----------
    MOVED  ACTIVE   AWAITING TRANSIT MOVING TO TRANSIT IN TRANSIT STATION
--------- -------   ---------------- ----------------- ------------------
       70      32           0                1                  2

VOL=SH20C0  MOVING TO TRANSIT IBM76421 TO EMC00547      28 DATASETS TO MOVE
VOL=SH20C3 IN TRANSIT STATION EMC00547 -- EMC00547       1 DATASETS TO MOVE
ACTIVE DATASETS FOR VOLUME SH20C3     NUMBER JOBS  JOBNAMES HOLDING ENQUEUE
------------------------------------- -----------  --------------------------
DB2.WORKFILE.VSH20C3.PS.A0001                 1   (JOB5)
VOL=SH20E0 IN TRANSIT STATION EMC00547 -- EMC00547       3 DATASETS TO MOVE
ACTIVE DATASETS FOR VOLUME SH20E0     NUMBER JOBS  JOBNAMES HOLDING ENQUEUE
------------------------------------- -----------  --------------------------
DB2.PAYROLL.VSH20E0.PO.A0002                  1   (CICS101)
DB2.PAYROLL.VSH20E0.VSAM.CLUSTER              2   (CICS101,DB2BATCH)
DB2.DSNDBC.DSNDB06.DSNKKX02.I0001.A001        2   (CICS101,DB2BATCH)
```

In the example above, the FDRPAS transit station job has completed moving the first two volumes to the transit stations and is still moving the third volume. Seventy (70) of the data sets on those two volumes were either inactive when the volume reached the transit station or became inactive sometime before this display was requested.

Since “Display DSN” is set to YES, it displays each data set that is still active. It also shows the number of jobs or tasks that hold the SYSDSN enqueue on the data set and lists the first few job names.

Once the display shows that all volumes are in transit stations, you see the active data sets that remain and the jobs or tasks that own them. You can then shut down those applications. FASTMOVE tests the enqueue every few seconds, and moves the data set with instant technology as soon as it becomes free.

For a MOVE job, the display is similar except that there is no transit station information. You can still see the data sets for which MOVE is waiting and decide when you will free them. As described in Section 325.5 “FDRMOVE Target Volume Initialization”, you can let a MOVE job run for a long time and it moves data sets (using normal I/O) as soon as they become free. You can monitor the remaining data sets and may choose to terminate the MOVE job and move the remaining data sets with FASTMOVE.

The display may also show “* CHECK ERROR MESSAGES *” if any error messages have occurred, or “* MOVE IS SUSPENDED *” if the job has been suspended.

NOTE: FDRMOVE processes all data sets selected from a given volume in parallel, so the data set count shown on the display does not change until FDRMOVE has finished all selected data sets on the volume.

325.11 FDRMOVE SPECIAL CONSIDERATIONS

SECURITY

By default, every data set moved by FDRMOVE invokes security checks from the z/OS system allocation and catalog functions. The user ID under which FDRMOVE is running **MUST** be authorized to create and update **ALL** selected data sets. The security system overhead may be significant if many data sets are involved. For this reason, the default security is **not recommended**.

To reduce this overhead and better control security, FDRMOVE supports a security STGADMIN profile that allows FDRMOVE to bypass security while moving data sets. To invoke this support:

- Specify the STGADMIN operand on the MOVE or FASTMOVE statement, for example,

```
FASTMOVE TYPE=DSF,STGADMIN,other operands
```

- Authorize the user ID under which FDRMOVE runs to profile `STGADMIN.ADR.STGADMIN.MOVE` in class FACILITY (any authority, including READ, is adequate). All known security systems support such profiles
- If the user ID is authorized to that profile, all security checks from all system components invoked by the FDRMOVE job is bypassed. It does not affect any other jobs

The advantages of STGADMIN are:

- Security overhead is reduced
- The user ID under which FDRMOVE runs is authorized to move any data set, but has no authority to those data sets outside of FDRMOVE. This may be a significant advantage if a third party contractor is running FDRMOVE at your installation.

RECOMMENDATION: ***INNOVATION DATA PROCESSING strongly recommends specifying the STGADMIN operand in FDRMOVE jobs to use the STGADMIN.ADR.STGADMIN.MOVE profile for all FDRMOVE operations.***

SECURITY SYSTEMS

It is possible that different security rules exist on various LPARs and that the security profiles for all the data sets that FDRMOVE selects do not exist on the LPAR where it is running. This causes security error messages unless STGADMIN is specified.

RECOMMENDATION: *INNOVATION DATA PROCESSING recommends implementing STGADMIN in order to bypass this issue.*

ALLOCATION CONTROL PACKAGES

Your installation may have a software package that controls or influences the volumes on which new DASD data sets get allocated. Such packages include:

- Allocation Control Center (ACC) from DTS Software
- BMC MainView SRM from BMC Software
- CA Allocate DASD Space and Placement from CA Technologies

There may be others.

If your installation's rules allow them to, these packages may override the selection of output volumes specified by the `NVOL=`, `NEWSTORGRP=`, and `ENEWSTORGRP=` operands of FDRMOVE. To allow the FDRMOVE control cards to be honored, you can either change the rules for the allocation control package, or add a special DD statement to the FDRMOVE JCL. The DD statement may be one of the following (or your installation may have specified a different ddname):

For ACC:

```
//ACCIGN   DD DUMMY
```

For MainView SRM (successor of POOL-DASD and ProSMS):

```
//PLDIGN   DD DUMMY
//PROIGN   DD DUMMY
//PROIGNOR DD DUMMY
//BYPASPRO DD DUMMY
```

For CA Allocate (successor of VAM and VDS):

```
//VAMBYPAS DD DUMMY
//VDSBYPAS DD DUMMY
```

UNMOVABLE TABLE

Certain data sets in your system may be active without a SYSDSN enqueue, so FDRMOVE cannot tell that they are active. Such data sets should not be moved. There may be other data sets that have DASD-location dependencies; they should also not be moved.

To make it easier to avoid moving such data sets, FDRMOVE supports an "unmovable table", a list of data sets that should not be moved. This unmovable table resides in the FDRMOVE program library. Actually, it is the same table that is used by COMPAKTOR (PGM=FDRCPK).

The unmovable table is converted into internally generated EXCLUDE statements for every FDRMOVE job and is shown in the control statement display.

The unmovable table distributed with FDRMOVE contains entries for:

- SYS1.VVDS.*
- SYS1.VTOCIX.*
- SYS1.LOGREC

You should identify additional data sets that are active without a SYSDSN enqueue, or that should not be moved for other reasons, such as:

- JES Procedure Libraries (PROCLIBs)
- JES SPOOL and Checkpoint data sets
- PAGE data sets (including PLPA and COMMON)
- Coupling data sets
- Tape Management System data sets
- LINKLIST Program Libraries
- CICS Journals
- SYS1.BRODCAST
- SYS1.MAN*x* (SMF) data sets
- Security System data sets
- FDRMOVE Program Library
- Non SMS-managed APF-Authorized Program Libraries
- Data sets used by programs specified with the NODSI option in the Program Properties Table (PPT) (PARMLIB member SCHED*xx*)

And add them to the table.

To update the table, go to the FDR ISPF main menu (see "Unmovable Table" in Section 325.11) and then:

1. Enter "I" (for Install)
2. Enter "5" to update the COMPAKTOR Unmovable Table (also used for FDRMOVE)
3. Make sure that the program library points to the FDRMOVE library and press ENTER
4. Now you can add or edit entries

You must have UPDATE authority to the FDRMOVE program library since the table must be stored in that library.

SYSTEM VOLUMES

Volumes in use by the operating system may have data sets that may not be enqueued and are continuously in use, such as:

- JES spool and checkpoint
- Page including PLPA and COMMON
- Coupling data sets
- LINKLIST data sets (If LINKLIST data sets are enqueued by LLA on your system, they do not need to be in the unmovable table)

You must either avoid moving the volumes containing these data sets, or add them to the unmovable table. Failure to do so may result in FDRMOVE moving the data sets, with possible system failure as the result.

Data sets that are indirectly cataloged (to a volume serial number of `******` for the IPL volume, or `&SYSRx` for an extended SYSRES volume) cannot be properly recataloged by FDRMOVE and should not be moved.

You must avoid moving these data sets and add them to the unmovable table or exclude these volumes from participating. However, you **CAN** move volumes containing these data sets non-disruptively with FDRPAS.

IBM RACF DATA SETS

FDRMOVE identifies one active RACF data set for the current system and automatically excludes it. However, if you have more than one data set in the active primary RACF data base, or you have an active backup RACF data base, or you have separate RACF data bases for different systems, you should add all of the active RACF data sets to the Unmovable Table, since RACF does not enqueue its data sets. You **CAN** move volumes containing active RACF data sets non-disruptively with FDRPAS.

TEMPORARY DATA SETS

FDRMOVE automatically excludes temporary data sets by generating this EXCLUDE:

```
EXCLUDE DSN=SYS+++++.T++++++.**
```

APF-AUTHORIZED LIBRARIES

APF-authorized program libraries are specified in the PROG*xx* member of PARMLIB. Non SMS-managed APF-authorized libraries must specify the volume on which they reside, so if such a library is moved by FDRMOVE, it is no longer APF-authorized and may cause program failures. You can move the volume containing a non SMS-managed APF-authorized library data set non-disruptively with FDRPAS, or you can update and activate a new PROG*xx* member after moving the library.

SMS-managed APF-authorized are not a problem; they are authorized on any volume.

CATALOGS

FDRMOVE cannot move ICF catalogs. Catalogs are automatically detected and excluded. A SIMMOVE warns you about all catalogs that cannot be moved.

To move a catalog, consult the IBM *z/OS DFSMS Managing Catalogs (SC26-7409)* manual. You may also want to read IBM informational APAR II13354 that has step-by-step instructions.

You can move a volume containing catalogs non-disruptively using FDRPAS.

SPECIAL DATA SETS

Special data sets on each volume, such as the VTOC, VTOCIX, VVDS, and ABR Model DSCB are automatically excluded by FDRMOVE. They are specific to the volume they reside on and never need to be moved.

NOTE: If you have changed the ABRINDEX value from the default of "FDRABR" in the FDR Global Options Table, ensure that you also change the ABRINDEX value to the same value in the FDRMOVE Options. Otherwise, ABR Model DSCBs are not determined as such and the entries are moved.

GENERATION DATA GROUPS (GDGS)

Generation Data Groups (GDGs) are automatically handled. There are no special considerations for GDGs.

CONNECTED CATALOGS (ALIASED CATALOGS)

Installation running with multiple LPARs. We have found that many installations have catalogs that are not connected (aliased) on every LPAR. This is usually an oversight but it may be done on purpose.

If FDRMOVE moves data sets on an LPAR where its catalog is not connected, FDRMOVE treats them as uncataloged data sets. VSAM data sets in these catalogs give an error message, while non-VSAM data sets are moved without updating any catalogs. Non-VSAM data sets allocated and moved to non SMS-managed volumes do not update the catalog. Non-VSAM data sets allocated and moved to SMS-managed volumes are cataloged using the catalog structure of the system where FDRMOVE is run.

Therefore, you must check all LPARs for this condition before running FDRMOVE.

IBM DFSMSHSM MIGRATION VOLUMES

DFSMShsm Migration volumes on DASD must be excluded from FDRMOVE since the DASD volume that the migrated data set resides on is required for DFSMShsm to properly locate the migrated image so the data set can be restored. However, FDRPAS can move the entire migration volume since the DASD volume serial number would remain the same.

UNCATALOGED DATA SETS

If some data sets are cataloged in a catalog that is not active on all systems, be sure to run FDRMOVE for those data sets on a system where they are cataloged. Otherwise, FDRMOVE treats them as uncataloged data sets. Non-VSAM data sets allocated to non SMS-managed volumes are moved without updating any catalogs, which may cause failures on systems where they are cataloged. Non-VSAM data sets allocated to SMS-managed volumes are moved and cataloged using the catalog structure of the system where FDRMOVE is run. Uncataloged VSAM clusters are not moved.

NOTE: Data sets may erroneously appear to be uncataloged if their alias is not connected to the proper user catalog on one or more systems.

Volumes that are copies of online volumes may appear to contain data sets that are not cataloged. These volumes should not be moved with FDRMOVE. They include

- Volumes created by DFSMSdss full-volume COPY without the COPYVOLID option
- Volumes created by IBM TDMF for backup purposes
- The original source volume of a IBM TDMF migration
- Volumes created by TSO FlashCopy commands (for example, FCESTABL)
- Volumes created by EMC SPLIT of a BCV, or EMCSNAP, if the target is online after the operation
- There may be other volumes of this type

FDREPORT can be used to identify uncataloged data sets. Member **MOVREP04** in the ICL library is an example of an FDREPORT job that identifies data sets that are not cataloged or cataloged to nonstandard catalog structures.

UNMOVABLE DATA SETS

PS, PO, and DA data sets on non SMS-managed volumes can be marked as unmovable in the Format 1 DSCB (via `DSORG=PSU`, `POU`, or `DAU`). An unmovable data set can ONLY be moved to the same track addresses on the output volume as it occupied on the input volume. An unmovable data set with more than three extents cannot be moved by FDRMOVE.

ARCHIVED DATA SETS

Data sets that are archived only have catalog entries that indicate their archive status. Since these data sets are not physically on DASD, FDRMOVE does not change the catalog entries for these data sets.

DATA SETS CATALOGED IN MULTIPLE CATALOGS

If you have separate master catalogs for your various systems, certain system data sets may be cataloged in each catalog. FDRMOVE can only update the catalog for the system it is running on, so it is the user's responsibility to update the other catalogs.

SMS CLASS CONSIDERATIONS

When an SMS-managed data set is moved, the SMS class names (data class, management class, and storage class) are retained, unless the user changes them with the `MGMTCLAS=` or `STORCLAS=` operands. The data class cannot be changed. The Automatic Class Selection (ACS) routines are not called, and the class names are not checked for validity.

FLASHCOPY AND EMCSNAP

FlashCopy and EMCSNAP do not support cascading relationships. In other words, if a data set on the source volume is currently the target of a Flash or Snap that has not completed, then Flash or Snap cannot be used to copy the data set to the FDRMOVE target volume. In this case, normal read/write I/O is used to copy the data sets.

If FASTMOVE used FDRPAS to move a volume to a transit station, this is not an issue unless new Flash or Snap has been issued since that move.

EMCSNAP is not used on data sets less than 15 tracks.

FLASHCOPY ON HDS DASD

Hitachi (HDS) has a restriction that FlashCopy cannot be used to a volume that is a PPRC (TrueCopy) primary volume. If you wish to use FASTMOVE to move data sets to such volumes, you must terminate the PPRC relationship before running FDRMOVE. If you do not, normal read/write I/O is used to copy the data sets.

HDS FlashCopy is not used on data sets less than 15 tracks.

Hitachi has a limitation on the number of active FlashCopy extents per volume, so if many data sets or extents must be copied, this limit may be exceeded. If the limit is exceeded it uses normal I/O to complete the move. To find out what your limit is, issue the TSO command `TSO FCQUERY DEV(uuuu)`; specify a device in the source control unit and also in the target control unit, whichever ones are HDS DASD. The output looks like this:

TSO FCQUERY DEV(17FC)

```
FCQUERY Formatted -2
DEVN SSID LSS CCA  CU    SERIAL       ACT    MAX XC PC CC RV  SEQNUM
17FC 9970  00  2C 2105 000000023006     52   3000  N  N  N  N 00000000
```

MAX is the maximum number of FlashCopy extents supported per volume, and ACT is the number currently active.

Hitachi has a limitation of on the total active FlashCopy extents per control unit, which is set by the HDS CE.

FDRINSTANT

If you are using FDRINSTANT to do full-volume backups of the FASTMOVE source volumes, you should be aware that FDRINSTANT might not work while a volume is in a transit station. This is because the volume will no longer be in the same DASD subsystem as your FDRINSTANT target devices. This also applies to other software that depends on fast replication facilities.

Obviously, if you intend to backup the new FDRMOVE target volumes, you need to setup new FDRINSTANT jobs for those volumes.

FASTMOVE PERFORMANCE

The data rate depends on the configuration, the number of FASTMOVE subtasks, the number of data sets to be moved, and other factors. FASTMOVE can typically move up to 1TB of data per minute, residing within no more than 250 data sets moving 16 volumes concurrently. FASTMOVE can typically allocate, catalog and move over 1000 small data sets per minute with 16 concurrent volumes. Moving 16 concurrent volumes requires 2 FDRMOVE jobs running concurrently.

Although FASTMOVE uses FlashCopy, EMCSNAP, or Snap to quickly move data sets from the transit stations to the target volumes, performance can be affected by a number of factors. FASTMOVE does not use a large amount of CPU time, but if sufficient resources are not available to the FASTMOVE task, it impacts performance.

It is highly recommended that the Indexed VTOC (VTOCIX) be active on all source and target volumes, in order to improve performance.

Since FDRMOVE must update the catalog for every data set or component moved, catalogs should be tuned for performance. See the IBM *z/OS DFSMS Managing Catalogs (SC26-7409)* manual for guidance.

TRANSIT STATIONS

FASTMOVE works best when the number of offline transit station devices equals or exceeds the number of input volumes. FASTMOVE is able to move all the input volumes to a transit station. Then, when you "bounce" the application that is using the data sets, all of them can be moved at once.

However, it may not be possible to provide a sufficient number of transit stations. FASTMOVE still works, but in stages. It moves input volumes to the transit stations until it runs out of stations. The FDRMOVE ISPF panel or the console STATUS command shows you what volumes are in transit stations and what data sets FASTMOVE is waiting for. Now when you bounce the application FASTMOVE moves the selected data sets from the transit stations.

Assuming that all selected data sets on that set of input volumes have been moved, they are swapped back to their original devices, and other volumes move to the transit stations. Now, you can bounce the application again to move the data sets from those volumes. This is repeated until all selected data sets have been moved.

IBM HYPERSWAP AND EMC AUTOSWAP

You cannot use FDRMOVE with **FASTMOVE TYPE=DSF** if IBM GDPS HyperSwap, IBM Basic HyperSwap, or EMC AutoSwap is active on the source DASD; HyperSwap and AutoSwap volumes fail in the FDRPAS TRANSIT step. However, FDRMOVE supports HyperSwap and AutoSwap volumes with **MOVE TYPE=DSF**.

CSA AND ECSA USAGE

FDRMOVE has no special CSA or ECSA usage, the same as any batch job.

The FDRPAS TYPE=TRANSIT job uses a small amount of CSA/ECSA while a volume is actively being moved to or from a transit station.

ENQUEUES AND RESERVES

FDRMOVE requires that SYSDSN enqueues (data set name enqueues) be propagated as global enqueues to all systems, so that it can determine what data sets are active. This is a standard procedure for all sites with shared DASD.

However, with multiple sysplexes or monoplexes, the SYSDSN enqueue must be propagated to all plexes where the data set may be in use.

FDRMOVE has no special consideration for hardware reserves. There is no requirement that all reserves be converted to global enqueues.

CA MII (CA MIM)

FDRMOVE does a conditional enqueue on SYSDSN for each data set to be moved. Since it does this as it is reading the VTOC of a source volume, it might enqueue many data sets in a very short period.

- If you are using CA MII (Multi-Image Integrity component of CA MIM), prior to release 11.6, this may result in a large number of contention messages from MII, which may flood and backup your console.
- To suppress these messages, see member FDRCONXT in the Installation Control Library (ICL) installed with FDRMOVE. This describes a CA MII exit that can be activated by a MIM console command to suppress those messages for FDR programs and it does not affect any other MIM operations.

CA MII does not support GQSCAN (except for the LPAR that FDRMOVE is executing on) so that FDRMOVE cannot determine the job names if it is owned on other system. If the data set is enqueued on this LPAR and also on other LPARs, FDRMOVE may display the local job names plus the MII address space name.

The following message is displayed by a SIMMOVE or a STATUS or ISPF monitor if the data set is not enqueued on this LPAR; it may or may not be enqueued on another system but FDRMOVE cannot determine its enqueue status.

```
FDR184        !---FOR MOVING DSN=DGIP.DEV.SSBA1FR.G0497V00     MIM-JOBS(UNKNOWN ON OTHER LPARS)
```

If you run a SIMMOVE on each LPAR, then you can determine which jobs are holding the enqueues.

MULTI-VOLUME DATA SETS

FDRMOVE must always move multi-volume data sets to the same number of volumes they currently occupy. When moving multi-volumes, the NVOL= list must always contain sufficient volumes to contain the data set. However, it is possible to move one or more pieces of a multi-volume data set without moving the rest.

You can run a SIMMOVE step that looks for multi-volume data sets. For each input volume, it displays the highest number of volumes (HIGHEST SEQ#). Take the highest number as the minimum number of NVOL= volumes to move these data sets successfully. This does not work for non SMS-managed VSAM clusters.

If you do not provide sufficient NVOL= volumes, then the FDRMOVE step gets an error message similar to this:

```
FDR156** ALLOCATE FAILED FOR 00001 TRK COMP=X'0004-041C0416' VOL=volser DSN=dsname
```

VOL= is the first volume that FDRMOVE tried. The message shows COMP=X'0014 if there was insufficient space on the first volume tried, and there was no other place to put the data sets on the NVOL volumes.

CANDIDATE VOLUMES

SMS-managed VSAM clusters with candidate volumes are handled automatically:

- If the SMS storage class is not guaranteed space, then candidate volumes are simply asterisks (*) in the catalog and do not need to be updated.
- If the SMS storage class is guaranteed space, then a “candidate space” is created on each candidate volume. FDRMOVE will move these candidate spaces if the volume is selected (CATDSN= or ALLDSN).

For SMS-managed non-VSAM data sets with candidate volumes are handled automatically:

- If the SMS storage class is not guaranteed space, then candidate volumes are simply asterisks (*) in the catalog and do not need to be updated.
- If the SMS storage class is guaranteed space, then a “candidate space” is created on each candidate volume. FDRMOVE moves these candidate spaces if the volume is selected (CATDSN= or ALLDSN).

Non SMS-managed VSAM clusters with candidate volumes are handled:

- Candidate volumes are marked as candidates in the catalog. FDRMOVE updates the candidate list if there are sufficient unused volumes in the NVOL= list, after all active pieces of the cluster are moved. If there are not sufficient unused volumes, some candidates are updated and some are not. If a cluster has the IMBED attribute (no longer supported by IBM) then candidates are not updated.

Non SMS-managed non-VSAM data sets:

- Specific candidate volsers (these are very rare) appear in the catalog when you provide extra volsers at allocation time. They are not marked in any special way by IBM and do not appear in the DASD VTOCs so FDRMOVE does not update the volsers.

For non SMS-managed volumes where FDRMOVE was not able update the candidate volsers, you need to locate these data sets and use the IDCAMS ALTER command with REMOVEVOLUMES to delete the candidates and ADDVOLUMES to add new candidates. Note that the number of candidate volumes required may be less on the new DASD volumes than on the old DASD volumes because the new DASD are so much larger; you may not need any candidates.

A SIMMOVE step with VTOCEMPTY=CHECK and SELECT CATDSN= with NVOL= (for all possible new volsers) identifies all pieces of multi-volume data set that have not been moved to those NVOLs, including candidate volsers for VSAM.

HIERARCHICAL FILE SYSTEM (HFS) AND ZSERIES FILE SYSTEM (ZFS) DATA SETS

FDRMOVE can move Hierarchical File System (HFS) and zSeries File System (zFS) data sets (used by UNIX System Services (USS)) as long as they are not active. FDRMOVE does not quiesce these data sets, so it is your responsibility to free the data sets if they must be moved. With FDRPAS, the entire volume these Hierarchical File System (HFS) and zSeries File System (zFS) data sets are on can be moved regardless if they are active or not.

DISABLENEW= OPTION OF FDRMOVE

If you specify the FDRMOVE option `DISABLENEW=YES`, then FDRMOVE sets all input SMS-managed volumes to a status of DISNEW and non SMS-managed volume to a mount status of PRIVATE. The purpose of `DISABLENEW=` is to prevent new allocations to the input volumes once the move starts, assuming that the user wants to completely empty the input volumes and direct new allocations to the output volumes. For SMS, the input and output volumes would typically be assigned to the same storage group, allowing the new allocations to naturally flow to the new volumes.

The SMS status of DISNEW is very effective for accomplishing this. However, there is a consideration. If a new SMS configuration is activated (the SMS "ACTIVATE" function), then all volumes are set to the SMS status indicated in the SCDS being activated. Since the FDRMOVE input volumes were originally marked as ACTIVE in the configuration, the ACTIVATE will probably return them to ACTIVE, allowing new allocations to go to those volumes.

Before you activate a new SMS configuration, during or after an FDRMOVE operation, you must set those input volumes to DISNEW in that configuration before activation. This is very important. You must communicate this to any group or person who may do an ACTIVATE.

Also, if you have separate SMS configurations for each LPAR, `DISABLENEW=YES` only disables the volumes on the current LPAR, so you are responsible for manually disabling the volumes on the other LPARs before moving any data sets.

NOTE: An IPL does not reset the volume status, only an ACTIVATE.

EXTENDED ADDRESS VOLUMES (EAVS)

Before z/OS V1R10, DASD storage was limited to 65,520 cylinders per volume. z/OS V1R10 supports Extended Address Volumes (EAVs), which are by definition 65,521 cylinders or larger. The maximum size of a volume is 262,668 cylinders with z/OS V1R10.

NOTE: In z/OS 1.13 and higher, an EAV volume can be up to 1 TB in size (1,182,006 cylinders).

The cylinder addresses greater than 65,535 on an Extended Address Volume (EAV) are referred to as the *Extended Address Space (EAS)*. These cylinder addresses are represented by 28-bit cylinder numbers. The cylinder addresses below 65,536 on an Extended Address Volume (EAV) are referred to as the *base addressing space*. These cylinder addresses are represented by 16-bit cylinder numbers or by 28-bit cylinder numbers whose high order 12 bits are zero.

The track address with 28-bit cylinder numbers has the following format: *track address* is a 32-bit number that identifies each track within a volume. It is *CCCCcccH* in the hexadecimal nibbles format where:

- "*CCCC*" is the low order 16 bits of the cylinder number,
- "*ccc*" is the high order 12 bits of the cylinder number, and
- "*H*" is the four-bit track number.

Extended address volumes are divided into *track-managed space* and *cylinder-managed space*.

- Track-managed space is the space on the volume that is managed in tracks and cylinders. Track managed space ends at cylinder address 65,519. Each data set occupies an integral multiple of tracks. Track-managed space also exists on all non-Extended Address Volumes (EAVs).
- Cylinder-managed space is the space on the volume that is managed only in Multi-Cylinder Units. A Multi-Cylinder Unit (MCU) is 21 cylinders. Cylinder-managed space begins at cylinder address 65,520. Each data set occupies an integral multiple of multi-cylinder units (MCUs). Space requests targeted for the cylinder-managed space are rounded up to the next multi-cylinder unit. Cylinder-managed space only exists on Extended Address Volumes (EAVs). An EAS-eligible data set is one that can be allocated anywhere on an EAV.

EAS-eligible data sets allocated on an EAV are created with Format 8 and Format 9 DSCBs in the VTOC. A Format 8 DSCB is equivalent to a Format 1 DSCB and the Format 9 DSCB provides additional attribute data and a set of pointers to possible Format 3 DSCBs.

For an Extended Address Volume (EAV), the system and storage group breakpoint value (BPV) helps direct DASD space requests to cylinder or track-managed space. The breakpoint value is expressed in cylinders. When the size of a DASD space request is the breakpoint value or greater, the system prefers to use cylinder-managed space for that extent. This rule applies to each request for primary or secondary space for data sets that are eligible for cylinder-managed space. If cylinder-managed space is insufficient, the system uses the track-managed space or uses both types of spaces. When the size of a DASD space request is less than the breakpoint value, the system prefers to use the track-managed space. If space is insufficient, the system uses the cylinder-managed space or uses both types of spaces. The breakpoint value (BPV) is specified in the IGDSMS*xx* PARMLIB member. This parameter is optional and if not specified, the default value of 10 cylinders is used. A breakpoint value can also be specified for each storage group and overrides any value in the IGDSMS*xx* PARMLIB member.

EAS-eligible data sets include the following:

- SMS-managed and non SMS-managed VSAM data sets
- zFS data sets
- Sequential data sets, including extended, basic, and large formats
- PDS and PDSE data sets
- Direct (BDAM) data sets

- Data set allocated with undefined DSORGs

Non-EAS eligible data sets include:

- HFS data sets
- PAGE data sets
- VTOC and VTOC index data sets
- VSAM data sets with IMBED or KEYRANGE attributes

EATTR=OPT CONSIDERATIONS

The FDRMOVE MOVE function supports `EATTR=OPT` allowing the MOVE process to allocate data sets in the cylinder-managed space of Extended Address Volumes (EAVs). The allocation of data sets in the cylinder-managed space by FDRMOVE follows these rules:

- For a non-VSAM data set, if the file has `EATTR=OPT` specified in the DSCB, or if the `EATTR=OPT` keyword is specified on the MOVE command and the file does not have `EATTR=NO` specified in the DSCB, the FDRMOVE MOVE function allows the allocation to go to cylinder-managed space.
  - A single-volume non-VSAM data set is allowed to round up to a multiple of 21 cylinders according to the normal allocation rules. That is, if the size is equal to or larger than the breakpoint value (BPV; default is 10 cylinders), the data set is allowed to be rounded up and allocated in cylinder-managed space.
  - A multi-volume non-VSAM data set is not allowed to round up. If the size is not a multiple of 21 cylinders, the allocation is allowed to be split between cylinder-managed space and track-managed space. If the size is less than 21 cylinders, the data set is allocated in track-managed space.
- VSAM clusters are eligible to be allocated in cylinder-managed space by default; `EATTR=OPT` does not need to be specified.
- If the part of the data component on the current volume is a multiple of 21 cylinders, it is allocated in cylinder-managed space.
- If the part of the data component on the current volume is not a multiple of 21 cylinders, and it is in more than one extent, the allocation is allowed to be split between cylinder-managed space and track-managed space. For example, a component with 22 cylinders may get 21 cylinders in cylinder-managed space and 1 cylinder in track-managed space.
- If the part of the data component on the current volume is not a multiple of 21 cylinders, and it is in one extent:
  - If the cluster is single-volume, and the component is larger than 105 cylinders, then it is rounded up to a multiple of 21 cylinders and allocated in cylinder-managed space. For example, a component with 106 cylinders is rounded up to 126 cylinders.
  - If the cluster is single-volume, and the component is smaller than 105 cylinders, then it is allocated in track-managed space with the same size.
  - If the cluster is multi-volume, then regardless of size, the component is allocated in track-managed space with the same size.
- Index components of VSAM clusters are always allocated in track-managed space with the same size unless the index component is a multiple of 21 cylinders.
- Clusters that have IMBED or KEYRANGEs are always allocated in track-managed space with the same size.

330 FDRERASE

330.0 INTRODUCTION

PURPOSE OF THIS MANUAL

The purpose of this manual is to provide you with the information to install, use, and understand FDRERASE.

WHAT IS FDRERASE?

FDRERASE is used to erase all data from DASD volumes before they are reused or removed from a data center.

FDRERASE is a proprietary program product of:

INNOVATION DATA PROCESSING
Innovation Plaza
275 Paterson Avenue
Little Falls, New Jersey 07424-1658

and is available for license exclusively from INNOVATION DATA PROCESSING

Reproduction of this User Manual is prohibited except by licensed users for their internal use.

A copy of this manual is available in Adobe PDF format on the INNOVATION DATA PROCESSING Documentation CD-ROM and on the INNOVATION DATA PROCESSING FTP site for online use and local printing.

© Copyright 2004, 2013 INNOVATION DATA PROCESSING

NOTE: The version number of FDRERASE always shows as 5.4/70 regardless of the version number printed in the documentation or the version number of FDRPAS and FDRMOVE.

USA CONTACTS

Corporate Headquarters

Voice	973-890-7300	Sales	sales@fdrinnovation.com
Fax:	973-890-7147	Support:	support@fdrinnovation.com
Home page:	http://fdr.com		

EUROPEAN OFFICES

France		Netherlands (& Nordic Countries)	
INNOVATION DATA PROCESSING		INNOVATION DATA PROCESSING	
191 Avenue Aristide Briand		Brouwerstraat 8	
94230 Cachan		1315 BP Almere	
Tel:	(33) 1 49 69 94 02	Tel:	036-534 1660
Fax:	(33) 1 49 69 90 98	Fax:	036-533 7308
frsales@fdrinnovation.com		nlsales@fdrinnovation.com	
frsupport@fdrinnovation.com		nlsupport@fdrinnovation.com	

Germany		United Kingdom	
INNOVATION DATA PROCESSING International Ltd.		INNOVATION DATA PROCESSING Ltd.	
Orleansstraße 4a		Clarendon House	
D-81669 München		125 Shenley Road	
		Borehamwood, Herts, WD6 1AG	
Tel:	089-489 0210	Tel:	0208-905 1266
Fax:	089-489 1355	Fax:	0208-905 1428
desales@fdrinnovation.com		uksales@fdrinnovation.com	
desupport@fdrinnovation.com		uksupport@fdrinnovation.com	

330.0 INTRODUCTION

CCEVS CERTIFICATION

FDRERASE V5.4 L50, the first z/OS secure erase utility to complete Common Criteria Evaluation and Validation Scheme (CCEVS) evaluation and Common Criteria EAL2 Augmented validation, has earned the right to display the International Common Criteria Recognition Arrangement (CCRA) certification mark on August 9, 2005.

FDRERASE V5.4 L50

Product Name:	**FDRERASE**
Technology Type:	Sensitive Data Protection
Date Certified:	August 5, 2005
Conformance Claim:	EAL2 Augmented ADV_SPM.1, ALC_FLR.2
Sponsor:	Innovation Data Processing
Phone:	973.890.7300
CC Testing Lab:	SAIC

FDRERASE is designed to comply with current U.S Government guidelines for erasing computer DASD prior to disposal… the Department of Defense concurring that the erasure of computer DASD prior to disposal, as specified in the ASD(C31) memo of June 4, 2001.

For common criteria EAL2 augmented evaluation Certificate to:
http://www.niap-ccevs.org/cc-scheme/st/index.cfm/vid/10064/maint/146

Validated Product

FDRERASE, Version 5.4, Level 70

Product Name: FDRERASE, Version 5.4, Level 70	**Key Words**: None
Product Type: Sensitive Data Protection	**Vendor**: INNOVATION DATA PROCESSING
Date: 23 January 2007	**POC**: Thomas J Meehan
Conformance Claim: EAL2 Augmented with ADV_SPM.1,ALC_FLR.2	**Phone**: 973-890-7300
PP Identifier: None	**Email**: tmeehan@fdrinnovation.com
Security Target:	**Web**: http://fdr.com/products/fdrerase/
Validation Report:	**CC Testing Lab**: SAIC Common Criteria Testing Laboratory
CC Certificate Image:	**Assurance Continuity Maintenance Report**:

PRODUCT DESCRIPTION

The TOE is an application that runs on a mainframe computer running the IBM z/OS operating systems. The TOE provides two different levels of DASD erasures. They are the ERASE and SECUREERASE functions. DASD erasures are actually performed by overwriting stored data to make the original data unrecoverable. This overwrite includes the Volume Table of Contents (VTOC), that is, the DASD directory. The TOE also provides a method to verify that user data has been erased. This is the VERIFY function.

The ERASE function overwrites every track of DASD with a track-length record, consisting of binary zeros by default. This single overwrite makes all data originally on each track unrecoverable by any normal system program running anywhere that has direct access to the DASD or through the DASD control unit. Original data, however, may still be recoverable through sophisticated laboratory techniques and special programs whose purpose is to recover data on DASD by commanding the DASD to skew read heads plus or minus a number of degrees. Any residual data recording on the “edge” of the track may be recoverable using such a technique.

The SECUREERASE function overwrites each DASD track a minimum of three times, writing a random pattern, a complement of the first pattern, and finally another random pattern, by default. This multiple overwrite process (optionally up to eight overwrites) makes the original data unrecoverable, even by sophisticated laboratory techniques applied to hard drives removed from the control unit.

The VERIFY function can be used to sample tracks on the erased volumes to ensure that they have been erased. By default it verifies a percentage of the volume but can verify the entire volume if needed.

330.0 INTRODUCTION

SECURITY EVALUATION SUMMARY

The evaluation was carried out in accordance with the Common Criteria Evaluation and Validation Scheme (CCEVS) process and scheme. The criteria against which the INNOVATION DATA PROCESSING, FDRERASE, Version 5.4, Level 50 TOE was judged are described in the Common Criteria for Information Technology Security Evaluation, Version 2.2 and International Interpretations effective on 28 January 2005. The evaluation methodology used by the evaluation team to conduct the evaluation is the Common Methodology for Information Technology Security Evaluation, Version 2.2, Revision 256, January 2004. Science Applications International Corporation (SAIC) determined that the evaluation assurance level (EAL) for the product is EAL 2 augmented with ADV_SPM.1 and ALC_FLR.2 family of assurance requirements. The product, when configured as specified in the INNOVATION DATA PROCESSING Software Distribution Process Description and Software Distribution Facility User Guide and the INNOVATION DATA PROCESSING FDRPAS and FDRERASE User Manual and Installation Guide, satisfies all of the security functional requirements stated in the INNOVATION DATA PROCESSING, FDRERASE Security Target, Version 1.0. One validator on behalf of the CCEVS Validation Body monitored the evaluation carried out by SAIC. The evaluation was completed in June 2005. Results of the evaluation can be found in the Common Criteria Evaluation and Validation Scheme Validation Report, (report number CCEVS-VR-05-0109, dated 5 August 2005) prepared by CCEVS.

ENVIRONMENTAL STRENGTHS

The TOE is a commercial product whose users require a low to moderate level of independently assured security. INNOVATION DATA PROCESSING, FDRERASE, Version 5.4, Level 50 is targeted at a relatively benign environment with good physical access security and competent TOE administrators and users. Within such environments, it is assumed that attackers will have a low attack potential. INNOVATION DATA PROCESSING, FDRERASE, Version 5.4, Level 50 supports the following five security functions:

1. Security Audit – The TOE writes to every track on the DASD in order to erase it. If this operation fails, the I/O is automatically retried by the DASD subsystem (hardware) and by standard IBM error recovery software in the operating system.

   If the TOE finds the DASD is not off-line, the TOE terminates with a non-zero completion code (return code) and outputs an error message with asterisks to the console and program listing indicating the erasure was incomplete, and the TOE makes no attempt to overwrite the data on that specific DASD volume.

2. User Data Protection – The TOE provides two DASD erasure functions: ERASE and SECUREERASE. Both functions overwrite DASD to ensure the risk of remaining residual data, if any, is commensurate with the risk of a person scavenging for user data. The ERASE function overwrites the DASD with one pass (or more, selectable by an input option, up to 8) of binary zero or of hexadecimal bytes chosen by the TOE user. The SECUREERASE function overwrites a DASD volume with a minimum of three passes (or more, selectable by an input option, up to 8) of hexadecimal bytes determined by the TOE.

   In addition, the TSF provides the VERIFY function to enable the TOE user to verify that physical tracks of the DASD have indeed been overwritten sufficiently that no residual information remains

3. Security Management – The TOE provides two DASD erasure options and identifies the DASD to be cleared.

   The TOE reports to the TOE user the outcome of a DASD overwrite, including: success; failure to access the DASD because the DASD is found to be on-line; and failure to overwrite a bad DASD track after successive attempts.

   The TOE provides the VERIFY function, to enable the user to verify that physical tracks of a DASD have indeed been overwritten sufficiently that no residual information remains.

4. Protection of Security Functions – The TOE protects against failure with loss of the secure state, which requires that the TOE preserve a secure state in the face of the identified failures. The TOE ensures that only DASD that has been varied off-line is available to the TOE. If it is not, the TOE does not attempt to overwrite the DASD and reports the failure to the TOE user. Also, the TOE checks before every write to see if the DASD has been varied online; if so, the operation is terminated with an error message.

   The TOE determines the manufacturer of the DASD before beginning to execute. This test is necessary since the external interface of the DASD for committing data to be written from a cache to the hard drive (termed “hardening”) varies by manufacturer, and the TOE has to determine the type and size of DASD it is attempting to overwrite.

   Throughout the process of performing a DASD overwrite, the TOE continually monitors for any I/O errors on the write and other I/O issued to the DASD. During an overwrite of a DASD, if twenty write errors are encountered, the TOE sends a message to the console and the TOE user identifying the DASD, and that the overwrite was a failure. The TOE then terminates and automatically returns to its inactive maintenance mode (that is, resident in the authorized library on DASD where it was originally installed).

5. Resource Utilization – The TOE notifies the user an operation did not complete in the event of identified failures. When a failure to write to a specific area of DASD occurs because of damage to the surface of the DASD, the TSF makes multiple attempts to write to the area in an attempt to overwrite any data that may reside there. If this fails, the TOE skips the affected area and continues with the overwrite until the complete DASD volume is overwritten.

330.1 FDRERASE (ERASING DASD)

INTRODUCTION

Your FDRPAS license may also include a license for FDRERASE, INNOVATION DATA PROCESSING's product for erasing data from DASD. FDRERASE can also be licensed separately.

FDRERASE is particularly useful for erasing all corporate data from the old FDRPAS source DASD after a successful swap to new DASD, especially if you are removing the old DASD subsystem. Failure to erase the old DASD may make your data available to the eventual new owner of the subsystem. This may be true even if you intend to scrap the old subsystem, unless you physically destroy the DASD within the subsystem.

FDRERASE can also be used anytime that you need to erase data from DASD, such as when you are reusing DASD for new purposes, or when you are leaving a disaster recovery site after a test or real disaster.

US Government rules (such as Health Insurance Portability and Accountability Act (HIPAA), Sarbanes-Oxley Act, Gramm-Leach-Bliley Act (GLB) and other government regulations (such as the European Data Protection Directive 95/46) and industry guidelines all require that various corporate data remain secure. Because of time constraints, a significant number of sites simply relabel old DASD volumes, which wipes out the VTOC but leaves the data records on the volume where they can be accessed by unauthorized users. FDRERASE can securely erase DASD volumes quickly and easily.

FDRERASE offers two levels of data erasure. All of these levels make your data unreadable through the DASD subsystem, using the DASD as z/OS count-key-data (CKD) DASD, and some levels make it unlikely that your data can be reconstructed even if the fixed-block architecture (FBA) DASD drives are removed from DASD subsystem:

- ERASE overwrites each track with a record of binary zeros. This is adequate to prevent most attempts to recover data from the internal fixed-block architecture (FBA) DASD drives. Optionally you can overwrite each track multiple times and can specify the data pattern to be used for each pass. There are also options to overwrite with a random pattern and to erase the track instead of overwriting.
- SECUREERASE overwrites each track three or more times, with varying patterns. This can be used for your most sensitive data, to ensure that it is unlikely to be recoverable from the internal fixed-block architecture (FBA) DASD drives.
- In addition, SIMERASE validates your control statements and confirms the DASD devices that are erased, without erasing anything. EMPTYVTOC empties the VTOC and VTOCIX without erasing data. VERIFY is used to verify that the DASD have been erased, and PRINT prints a summary of track contents on erased or unerased DASD.

Section 330.5 "FDRERASE Considerations" has more detail on these erase functions and the threats to your data that each one addresses.

FDRERASE operates only on offline DASD, such as the old source DASD of an FDRPAS swap. By default, it checks internal indicators to ensure that only FDRPAS source DASD or empty DASD (no data files) are erased, but you can tell it to erase any offline DASD. There is an option to select online volumes and vary them offline for erasing.

FDRERASE can erase a total of 8190 DASD volumes in a single step, processing up to 64 DASD volumes concurrently.

SUPPORTED SOFTWARE AND HARDWARE

FDRERASE executes on any release of z/OS or OS/390.

FDRERASE supports a wide variety of DASD devices from hardware vendors including IBM, EMC, Oracle StorageTek, and Hitachi.

FDRERASE supports DASD attached via FICON, ESCON, and parallel (bus/tag) channels.

USER EXPERIENCES

"We are a government site that has sensitive data that must be erased after a DR test. A minimal INIT is not acceptable to our auditors as it only wipes out the VTOC and leaves the data on the DASD volume. At our latest DR test, we ran 6 jobs (one for each of the 6 storage frames) and they completed in 2.5 hours. The total amount of storage was 13 terabytes of data, which was comprised of (96) 3390 mod 3's, (1200) 3390 mod 9's, and (96) 3390 mod 27's."

LIMITING ACCESS TO FDRERASE

Because FDRERASE can destroy data if misused, you should limit access to its functions. You can use your security system to limit access to the program library containing FDRERASE or the program name FDRERASE. FDRERASE also performs internal security checks that limit access to specific functions (see "Securing FDRERASE" in Section 330.5).

PRODUCT DEMO

Product demos are available on the INNOVATION DATA PROCESSING web site for FDRERASE. Go to website http://fdr.com and click the FDRERASE link under the "DEMOS" icon.

The direct link for the FDRERASE Product Demo is: http://www.fdr.com/demo.cfm

330.2 FDRERASE CHECKLIST

This is a simple checklist of steps when preparing to erase DASD with FDRERASE. Please review the rest of the FDRERASE documentation to understand the steps in this checklist.

- ❒ Review Section 330.5 "FDRERASE Considerations" to understand the different types of erase (and choose the appropriate one to fit your need) and to understand the impact of erase on the type of DASD you are erasing.
- ❒ At your home site, you probably want to restrict which users can use FDRERASE functions. (See "Securing FDRERASE" in Section 330.5.)
- ❒ FDRERASE obviously has the capability of erasing data that should not be erased. Although FDRERASE has checks to try to ensure that only appropriate DASD are erased, these can be circumvented. **Only you can positively identify the DASD that can be safely erased, and it is your responsibility to do so.** You should do the same sort of checks that you would do before initializing a DASD volume with ICKDSF (for example, commands INIT, INSPECT, REFORMAT, TRKFMT).
- ❒ FDRERASE only operates on DASD that are offline to the system where it is executing. Except on most EMC subsystems, FDRERASE cannot tell if a DASD volume is online to another system. Even the ONLINE=VARYOFF operand (which varies online DASD offline for erasing) only operates on the current system. It is your responsibility to ensure that the DASD are offline on all systems. The console command "`D U,,,`*uuuu*`,`*nn*" can quickly tell if a range of DASD volumes are offline; execute this on each system before using FDRERASE. If your systems are in a sysplex, the console command "`ROUTE *ALL,D U,,,`*uuuu*`,`*nn*" executes this on every system. For example, the following shows that 01C1 is offline to all systems but 01C0 is online (O) to 2 out of 3 systems:

ROUTE *ALL,D U,,,1C0,2

```
ROUTE *all,d u,,,1c0,2
CPUA     RESPONSES ----------------------------
IEE457I 14.02.46 UNIT STATUS
UNIT TYPE STATUS        VOLSER     VOLSTATE
01C0 3390 OFFLINE                      /RSDNT
01C1 3390 OFFLINE                      /RSDNT
CPUB     RESPONSES ----------------------------
IEE457I 14.02.46 UNIT STATUS
UNIT TYPE STATUS        VOLSER     VOLSTATE
01C0 3390 O             SYM000     PRIV/RSDNT
01C1 3390 OFFLINE                      /RSDNT
CPUC     RESPONSES ----------------------------
IEE457I 14.02.46 UNIT STATUS
UNIT TYPE STATUS        VOLSER     VOLSTATE
01C0 3390 O             SYM000     PRIV/RSDNT
01C1 3390 OFFLINE                      /RSDNT
```

- ❒ The default of CHECKTARGET=YES to erase only empty DASD (no data sets) and those with no valid volume label (such as FDRPAS source DASD) is usually appropriate; override it only if you are sure that the non-empty DASD are to be erased. However, CHECKTARGET=NO may be appropriate at a disaster site, where you are erasing your data from DASD before you leave the site.
- ❒ Likewise, ONLINE=VARYOFF, to vary DASD offline before erasing them, is usually appropriate only at a disaster site, to vary your volumes offline and erase them.
- ❒ You get the best performance from FDRERASE if you specify many DASD volumes to be erased in one FDRERASE job, on one MOUNT statement, letting FDRERASE manage the DASD that FDRERASE erases concurrently. If the

specified DASD volumes are in multiple subsystems or logical control units, FDRERASE is able to manage the erase tasks dynamically, providing a good elapsed time while avoiding overloading any one subsystem or the physical FBA DASD volumes with the subsystem.

- ❒ You can use the VERIFY or PRINT functions of FDRERASE to confirm that data has been erased.
- ❒ To meet auditing requirements, you may need to keep the job listings from FDRERASE. They serve as records of the DASD that were erased.

330.3 FDRERASE JOB CONTROL REQUIREMENTS

To execute FDRERASE, the following JCL statements are required. Please review the examples in Section 330.6 "FDRERASE Examples" for a better understanding of the JCL requirements of FDRERASE.

EXEC STATEMENT

Specifies the program name (`PGM=FDRERASE`), region requirement (`REGION=0M` is recommended), and optional `PARM=` field that may contain the first FDRERASE control statement. For example,

```
//SWAP     EXEC PGM=FDRERASE,REGION=0M,
//              PARM='ERASE TYPE=FULL'
```

The parameter (PARM) may also contain a MOUNT control statement, separated from the first by a slash (/), for example,

```
PARM='ERASE TYPE=FULL/MOUNT ERASEUNIT=17F4'
```

NOTE: There must be no space immediately before the slash. Additional control statements, if any, must be contained in the SYSIN data set.

STEPLIB OR JOBLIB DD STATEMENT

Specifies the load library where FDRERASE resides. The library must be authorized.

SYSPRINT DD STATEMENT

Specifies the output message data set; it is required. It is usually a SYSOUT data set but if it is assigned to a data set on tape or DASD, this DD statement must specify `DISP=MOD`. DCB characteristics are `RECFM=FBA` and `LRECL=121`; the block size defaults to 1210 on DASD or tape.

SYSPRTxx DD STATEMENT

Specifies an output message data set for an internal erase subtask. They are optional. If needed, they are dynamically allocated as `SYSOUT=*`, so you need to specify them only if you need to direct those messages elsewhere. It is usually a SYSOUT data set but if it is assigned to a data set on tape or DASD, this DD statement must specify `DISP=MOD`. DCB characteristics are `RECFM=FBA` and `LRECL=121`; the block size defaults to 1210 on DASD or tape.

FDRSUMM DD STATEMENT

Specifies the optional output message data set for an FDRERASE summary report. It is usually a SYSOUT data set but if it is assigned to a data set on tape or DASD, this DD statement must specify `DISP=MOD`. DCB characteristics are `RECFM=FBA` and `LRECL=121`; the block size defaults to 1210 on DASD or tape. It contains a one-line summary for each volume processed.

SYSUDUMP DD STATEMENT

Specifies the ABEND dump data set. Usually specifies a SYSOUT data set. Although not required, we strongly urge you to always include this DD statement, so that we can help you diagnose error conditions. If you have a debugging aid product on your system that would prevent the desired dump, please add the appropriate one of these statements to the JCL so that a fully-formatted dump is produced.

```
//ABNLIGNR DD DUMMY    Turn off Abend-Aid
//CAOESTOP DD DUMMY    Turn off CA OPT II & CA SYMDUMP
//DMBENAN  DD DUMMY    Turn off DumpMaster
//ESPYIBM  DD DUMMY    Turn off Eye-Spy
//IDIOFF   DD DUMMY    Turn off IBM Fault Analyzer
```

SYSIN DD STATEMENT

Specifies the control statement data set. Usually an input stream or DD * data set. It may be DUMMY if all necessary control statements are included in the `PARM=` on the EXEC statement.

IXSYSPRx DD STATEMENT

If the VTOC is to be rebuilt on any volume, this is used for messages from ICKDSF. "*x*" may be 0 to 9. If omitted, FDRERASE dynamically allocates it to SYSOUT=*.

IXSYSINx DD STATEMENT

If the VTOC is to be rebuilt on any volume, this is used to hold the control statements for ICKDSF. It should be a one-track DASD data set. "*x*" may be 0 to 9. If omitted, FDRERASE dynamically allocates it using `UNIT=SYSALLDA` as a temporary data set.

330.4 FDRERASE CONTROL STATEMENTS

The control statements consist of a main statement (ERASE, SECUREERASE, SIMERASE, EMPTYVTOC, VERIFY, or PRINT) and one or more MOUNT statements specifying the DASD devices to be processed.

EMPTYVTOC, ERASE, SECUREERASE, AND SIMERASE STATEMENT SYNTAX

The syntax of the EMPTYVTOC, ERASE, SECUREERASE, and SIMVTOC statements are:

```
EMPTYVTOC     TYPE=FULL
ERASE
SECUREERASE
SIMERASE
              ,ACTIVETARGET=BYPASS|          ,LOGMESS=NO|
                             PROCESS                  YES
              ,ALREADYERASED=BYPASS|         ,MAXEU=8|
                             PROCESS                  nn
              ,CHECKTARGET=NO|               ,MAXTASKS=64|
                           YES                        nn
              ,CONFERASE=NO|                 ,ONLINE=BYPASS|
                         YES                          VARYOFF
              ,CPYVOLID=NO|                  ,PRTDEFAULTS
                        YES                  ,SELTERR=NO|
              ,ERASEPASS=1|                            YES
                         nn                  ,VARYON=AFTER|
              ,ERASEPATTERN=hh...                      NOAFTER
              ,HARDENTIMEOUT=120|            ,VOLSORT=NO|
                             nnnn                     YES
```

PRINT AND VERIFY STATEMENT SYNTAX

The syntax of the PRINT and VERIFY statements are:

```
PRINT         TYPE=FULL
VERIFY
              ,ERASEDATA=NONE                ,ERASESTARTCYL=0|
                                                           ccccc
              ,ERASENUMB=100|                ,ERASESTARTTRK=tt
                         nnnnn
              ,ERASESKIP=15|                 ,ONLINE=BYPASS|
                         nnnnn                        VERIFY
```

EMPTYVTOC STATEMENT

EMPTYVTOC does not erase any data. It quickly writes the VTOC as empty (no data sets). EMPTYVTOC is a quick way to delete all data sets from a volume, or to initialize a volume. Many DASD can be initialized with simple control statements; up to 10 DASD volumes are processed in parallel. The VTOC size and location can be specified on the following MOUNT statements or it defaults to the existing VTOC size and location if it exists. The VTOCIX (indexed VTOC) is also created for a new VTOC or if the VTOCIX previously existed.

By default, EMPTYVTOC operates on offline volumes only. If you add the `ONLINE=VARYOFF` operand, the volume is varied offline (if it is not in use) and emptied. If you add the operand `VARYON=AFTER`, the volume is varied online after it is emptied.

WARNING: ***EMPTYVTOC does not erase any data, nor does it uncatalog data sets that were in the VTOC.***

ERASE STATEMENT

By default, ERASE overwrites every track on the selected DASD devices with a single track-length record consisting of binary zeros, obliterating all data records that previously existed on the track. You can optionally request that every track be overwritten multiple times (ERASEPASS=*n*) and you can optionally specify the value to be used in each byte of the overwriting record (ERASEPATTERN=), varying it from pass to pass. Special pattern values are used to request a random pattern, or a "quick-erase" pass (erasing the tracks instead of overwriting). If multiple passes are requested, it processes every track on the volume in each pass from the "top" of the volume to the beginning.

When the overwriting record is all binary zeros, the ERASE function is quite fast since very little data must be sent down the channel for each track, allowing many DASD to be erased in parallel. Other values may require more time since the entire record must be transmitted for each track. It waits at the end of each pass to be sure that the data is hardened (written) to the DASD before continuing; this may take some seconds.

ERASE is often adequate for erasing DASD that are sold, scrapped, or returned to the manufacturer, since it makes it difficult to recover the original data even if the hard drives are removed, more so if multiple passes and patterns are used.

PRINT STATEMENT

The PRINT statement prints information about the contents of selected tracks from an offline or online DASD. This can be used to verify that a DASD volume was actually erased. You can also use it to print the contents of a DASD volume before the erase.

By default, PRINT prints the first track from each of the first 100 cylinders on each selected DASD device, but you can specify operands to control the tracks printed. The operands for PRINT are listed at the end of the operand list that follows.

The print function prints one FDR175 message for each record on the selected tracks, including record 0 (R0) that is present on every track. However, if ERASEDATA=NONE is specified, only R0 is printed plus a count of the additional records. DASD erased with ERASE or SECUREERASE contain one track-length data record with the same value in every byte (unless ERASEPATTERN=FE was used); the value is the pattern used in the last erase pass on the track. See the description of message FDR175 for details of the print format.

SECUREERASE STATEMENT

SECUREERASE is essentially an ERASE operation, except that minimum of ERASEPASS=3 is forced and special patterns are used. The first pattern is a random value (generated values "00", "01", "FE", and "FF" are changed to "AA"), the second pass uses the complement of the first pattern, and the third pattern is a new random value (different from the first two). Up to 32 passes can be requested. The fourth pattern is a complement of the third. If more than four passes are requested, passes five through 32 use patterns of random values; instead of using the same value in every byte, each byte in each record in a cylinder is randomly generated; this is the processing described below for ERASEPATTERN=FE. The "odd" pass pattern is randomly generated, and the "even" pass pattern is the complement of the previous pass.

SECUREERASE can be used on your most sensitive data to make it very unlikely that the data could be recovered if the hard drives were removed. This may be necessary to meet legal requirements, such as for the protection of financial data, social security numbers, and other personal data.

SIMERASE STATEMENT

SIMERASE does not erase any DASD, but it can be used to validate your FDRERASE control statements, and to verify which DASD devices are erased once SIMERASE is changed to one of the other options. SIMERASE lists all DASD devices that meet the selection criteria. If you specified CHECKTARGET=YES (or let it default), SIMERASE verifies that the selected volumes are either empty or have no valid volume label. If you specified ONLINE=VARYOFF, it identifies the volume that will be varied offline (although it cannot guarantee that the VARY OFFLINE will work).

VERIFY STATEMENT

The VERIFY statement can be used to verify that a DASD volume has actually been erased. It reads the specified tracks on the selected volumes, and analyzes their contents to see it appears that they were erased with FDRERASE. To pass the verification, a track must contain a single full-track record containing the same character in every byte or containing the random pattern generated by ERASEPATTERN=FE.

If every verified track appears to be erased, the VERIFY reports that the volume is erased. Tracks that do not verify are printed in the format used by the PRINT statement (message FDR175) so that you can see what they contain. If more than ERASENUMB= tracks fail verification, VERIFY stops processing the volume.

By default, VERIFY verifies the last track from every cylinder on each selected DASD device, but you can specify operands to control the tracks verified. However, the elapsed time of the VERIFY is proportional to the number of tracks verified. The operands for VERIFY are listed at the end of the operand list that follows.

If all tracks pass verification, message FDR177 is printed to indicate this and to show what erase pattern was found (ERASE if the tracks contain no records, "*xx*" if all bytes contain X'*xx*' and "FE (Random)" if the FE pattern was detected.

Note that if you used operands to rebuild the VTOC after the volume was erased, the VTOC and VTOCIX tracks do not pass verification. You can successfully verify such volumes only if you use VERIFY operands such as ERASESTARTCYL= to start the verification after the VTOC and VTOCIX. If you plan to verify erased volumes, we suggest that you use the EMPTYVTOC function to rebuild the VTOC after running VERIFY.

EMPTYVTOC, ERASE, SECUREERASE, SIMERASE PRINT & VERIFY STATEMENT OPERANDS

TYPE=FULL

Must be specified on this statement.

ACTIVETARGET=

BYPASS – DASD that have active PPRC or EMC SRDF mirrors are not erased; you must terminate the PPRC/SRDF session before erasing the DASD. In addition, DASD in EMC subsystems that appear to be online to other systems are not erased.

PROCESS – DASD that have active PPRC or EMC SRDF mirrors are erased. This may slow down the erase since all the erased tracks must be mirrored to the remote DASD. In addition, DASD in EMC subsystems that appear to be online to other systems are erased.

Default: BYPASS.

NOTE: When ACTIVETARGET=PROCESS is specified and a PPRC or SRDF session is active, FDRERASE performance may be degraded.

ALREADYERASED=

BYPASS – DASD that have been previously erased by FDRERASE are bypassed with an explanatory message. This is useful if you need to restart an FDRERASE job so that volumes previously completed are not erased again. However, if it was previously erased with the CPYVOLID=YES operand in effect, which leaves the volume in a usable state, FDRERASE erases it again.

PROCESS – DASD that have been previously erased by FDRERASE are erased again.

Default: BYPASS.

CHECKTARGET=

NO – The contents of the offline DASD is not checked. Any offline volume can be erased. `CHECKTARGET=NO` is required if you wish to erase DASD other than empty volumes and those created by FDRPAS. There may be additional security requirements for this operand. (See "Securing FDRERASE" in Section 330.5.) If there are data sets on the volume, FDRERASE does not uncatalog them. You can make `CHECKTARGET=NO` invalid by setting option CHKTARGNO to "NO" in the FDR Global Options Table in the FDRERASE load library (see "CHKTARGNO" in Section 380.16).

YES – FDRERASE checks the DASD device before beginning an erase operation to ensure that the DASD is empty or does not have a valid volume label "VOL1". If the volume label contains "VOL1", it checks to be sure that the volume contains only a VTOC, VTOCIX, VVDS, and/or ABR Model DSCB. DASD with no valid volume label include FDRPAS SWAP source DASD and SWAPDUMP target DASD (these have "FDR3" instead of "VOL1"), as well as FDRINSTANT target DASD (these have "FDR1" instead of "VOL1".) Using `CHECKTARGET=YES` accepts all DASD whose label is invalid for any reason plus those with no volume label at all. If it fails these tests, the volume is bypassed with an explanatory message.

Default: YES.

WARNING: ***FDRERASE cannot ensure that the selected devices are offline to all other sharing systems. You must take great care not to select DASD that contain data that is currently in use or that is needed later as it is possible to erase a DASD volume that is in use on another system when especially when specifying the `CHECKTARGET=NO` operand.***

CONFERASE=

NO – Suppresses the messages and begins the erase immediately.

YES – Before beginning the erase, FDRPAS requests confirmation via console WTOR message FDRW01 that must be replied before the erase can start. All devices selected by each MOUNT statement are listed (note that some of them may be later bypassed because they are not eligible for erase). The FDRW01 message is preceded by FDR235 WTO messages defining the DASD volumes to be erased. Only one reply is required per FDRERASE step.

Default: NO.

CPYVOLID=

NO – After completing the erase, there is no VTOC on the volume. All tracks are erased; the label track (cylinder 0 track 0) is rewritten with a volume label with an ID of "FDR5" so that the volume cannot be varied online; the volume serial is the original volume serial of the volume. If you wish to use the volume, you need to initialize it with EMPTYVTOC or ICKDSF.

YES – If the volume has a valid volume label and VTOC before the erase, then FDRERASE invokes ICKDSF to rebuild an empty VTOC (and possibly a VTOCIX) after the volume is erased. If the volume was SMS-managed, the SMS flag is still on in the VTOC. The volume label has an ID of VOL1 so that the volume can be varied online.

By default, the volume has its original volume serial number, and the VTOC (and VTOCIX, if present) has its original location and size. You can specify options on the MOUNT statement to change the volume serial number, and to change the size and/or location of the VTOC.

Default: NO. `CPYVOLID=YES` is assumed if you specify `VARYON=AFTER`, EMPTYVTOC, or CHANGEVOL=.

NOTE: If you are erasing a large number of volumes, it is faster to use `CPYVOLID=NO` and run EMPTYVTOC in a separate job to re-initialize all of the volumes after all the ERASEs are complete.

ERASEPASS=

***nn* –** Specifies the number of times that FDRERASE overwrites each track. Note that FDRERASE starts at the top (highest cylinder) of each volume and continues through cylinder 0, then repeats for the next pass. At the end of each pass, FDRERASE ensures that the data has been written from cache to the back-end DASD. The overwriting data is a single full-track record; it is all binary zeros unless `ERASEPATTERN=` or SECUREERASE is specified.

If the function is ERASE, only values from 1 to 8 are accepted.

If the function is SECUREERASE, only values from 3 to 32 are accepted.

Default: 1 for ERASE and 3 for SECUREERASE.

ERASEPATTERN=

***hh…* –** Specifies the byte to be used for each erasing record used by the ERASE function (it is ignored for SECUREERASE). It consists of a series of bytes (up to eight bytes, two hex digits per byte). The first byte is used to fill the record for the first pass, the second for the second pass, and so on. If you specify fewer bytes than the value of `ERASEPASS=n`, binary zeros are used for the extra passes.

There are some pattern bytes with special meanings:

01 – Indicates that this pass is to erase the track instead of writing a pattern; 01 could be used as the last pattern to cause the track to be erased, leaving no records on the track.

FE – Indicates that random values are to be used. Unlike other patterns, the bytes throughout the record on each track varies, and a different pattern is used for each track within a cylinder. If "FE" appears twice in a row in ERASEPATTERN, the first pattern is randomly generated, but the pattern for the next pass is the complement of the preceding pass. If "FE" appears three times in a row, the third pattern is randomly generated, while a fourth occurrence of "FE" in a row is the complement of the third pattern, and so on. If any other pattern character appears between occurrences of "FE", the next "FE" generates a new random pattern.

For SECUREERASE, the patterns are random, as described earlier. You cannot override the patterns for SECUREERASE, although you can override the number of passes (`ERASEPASS=`).

Default: 00 for ERASE, random from SECUREERASE.

HARDENTIMEOUT=

***nnnn* –** Specifies the number of seconds (0-2550) that FDRERASE waits for data to be hardened (written from cache to DASD) at the end of each pass of ERASE and SECUREERASE. A value of 0 disables the hardening commands; 0 is valid only for ERASE, not SECUREERASE. On EMC subsystems, a value other the default of 120 causes FDRERASE to wait for the entire time out value even if the count of unhardened tracks stalls (stops decreasing). On other than EMC subsystems, the control unit continues to harden data even if FDRERASE has stopped waiting for it.

Default: 120 (2 minutes).

LOGMESS=

NO – No SYSLOG/console messages are written.

YES – Messages are written to SYSLOG (and usually to an operator console) documenting that the erase of each volume has completed (the "successful" message and all error messages).

Default: YES.

MAXEU=

nn – Specifies the maximum number of volumes (1-64) that can be erased concurrently if they reside in the same underlying FBA physical DASD or RAID group in the DASD subsystem, if it can be determined. This avoids performance degradation that can occur if many logical volumes that reside on the same underlying DASD are erased concurrently, because of contention for the disk heads and data paths to the DASD.

If a MOUNT statement in this FDRERASE step specifies DASD devices that reside on various underlying DASD or even in different DASD subsystems, then the MAXEU= and MAXTASKS= operands interact to control the active erase tasks. FDRERASE attempts to start up to MAXTASKS= erase tasks, as long as no more than MAXEU active tasks are directed to the same underlying DASD or RAID group.

FDRERASE uses queries that vary by DASD manufacturer to identify the underlying physical DASD or RAID group for each DASD volume, but it cannot determine the underlying DASD volumes in all DASD subsystems. If it cannot make this determination for some or all DASD specified, it does not limit the number of erase tasks affecting those DASD volumes.

You may be able to increase the value of MAXEU if the DASD are in a high-performance subsystem, but you may need to experiment to find an appropriate value.

MAXEU= is ignored if VOLSORT=NO is specified.

Default: 8

MAXTASKS=

nn – Specifies the maximum number of volumes that can be erased concurrently, from 1 to 64. If one of the following MOUNT statements selects a number of offline DASD volumes that exceeds MAXTASKS=*nn*, FDRERASE starts the indicated number of internal erase tasks; as each one finishes another one is started, until all selected DASD have been erased. The number of tasks can also be modified dynamically while FDRERASE is running. (See "Console Commands for FDRERASE" in Section 330.5.)

Note that MAXTASKS=64 (the default) requires a below-the-line private area of up to 8.5MB. If the available below-the-line private area on the system where FDRERASE is run is smaller, MAXTASKS= is automatically reduced, to avoid storage shortage errors, unless you have specified the MAXTASKS= operand.

MAXTASKS= applies to each MOUNT statement separately. The DASD volumes specified on each MOUNT statement are completely processed (up to MAXTASKS= concurrently) until they are all complete. Then the next MOUNT statement is processed.

If VOLSORT=YES is specified or defaulted, then the maximum concurrent erase tasks may be less than 64 because of the processing described above under MAXEU=, where FDRERASE attempts to limit the number of concurrent erases that affect the same underlying physical DASD or RAID groups. If the underlying physical DASD cannot be determined, but the total number of DASD to be erased exceeds MAXTASKS=, then FDRERASE spreads the active tasks by z/OS device address, to attempt to achieve the same result.

For EMPTYVTOC, the maximum value for MAXTASKS= is 10.

Default: 64 except EMPTYVTOC defaults to 10.

ONLINE=

BYPASS – DASD devices that are online are bypassed with an explanatory message. If the bypassed DASD was selected by a full 4-digit device address (not a prefix), the message causes the step to end with return code to call attention to the bypassed device.

VARYOFF – DASD devices specified on each MOUNT statement that are online to this system are varied offline and erased if they are not currently allocated to any task on this system. To ensure that you are varying the proper devices offline, WTOR message FDRW01 is issued to the operator's console to confirm each device.

Default: BYPASS.

WARNING: ***It is your responsibility to ensure that the online volumes that you have specified are not online or in use on any other system. If you erase a volume in use on another system, the results are unpredictable, and data is lost. ONLINE=VARYOFF is primarily intended for use when erasing DASD as you are leaving a disaster/recovery site since it relieves you of the need to manually vary all the volumes offline.***

NOTE: If you specify ONLINE=VARYOFF and the volumes to be varied offline contain data sets, you must also specify CHECKTARGET=NO in order to erase those volumes. If the volumes are empty, then CHECKTARGET=NO is not required.

PRTDEFAULTS

If specified, it lists the defaults for various operands that affect FDRERASE operation. If overriding operands are present before the PRTDEFAULTS operand, it displays those overrides. Overriding operands that follow the PRTDEFAULTS operand are not displayed.

SELTERR=

Specifies what happens at step termination if FDRERASE finds that one or more selected offline devices was not erased because it did not meet all the criteria, such as it was already erased (unless ALREADYERASED=PROCESS was specified) or not created by FDRPAS or empty (unless CHECKTARGET=NO was specified).

NO – A condition code or ABEND is not to be issued at step termination. You might use SELTERR=NO when you expect some devices in your ERASEUNIT= list may not be eligible.

YES – A condition code or ABEND is issued at step termination to call attention to the volumes that were bypassed.

Default: YES, unless overridden in the FDR Global Options Table (see "SELTERR" in Section 380.15).

VARYON=

AFTER – After erasing the DASD, the volume is varied online and remounted. CPYVOLID=YES is forced. The volume serial of the DASD must not match the serial of any other online volume (such as the FDRPAS target device if you are erasing an FDRPAS source DASD); if it does match you can use the CHANGEVOL= operand of the MOUNT statement to relabel the volume. The VARY ONLINE is done after the erase is complete and the FDR241 ERASE COMPLETE message is printed.

NOAFTER – The DASD volume is left offline after the erase.

Default: NOAFTER.

VOLSORT=

NO – The underlying DASD volumes are not used to sort or limit the DASD volumes erased concurrently. DASD are selected in the order that their UCBs are found in the operating system and up to MAXTASKS= DASD volumes are erased concurrently.

YES – FDRERASE uses hardware queries (that vary by DASD manufacturer) to identify the underlying physical FBA DASD or RAID group for each logical DASD selected for erase and the DASD volumes are sorted by that DASD ID. See "MAXEU=" in Section 330.4 to understand how this ID affects FDRERASE operation.

Default: YES.

ADDITIONAL PRINT AND VERIFY STATEMENT OPERANDS

The following operands can be specified ONLY on a VERIFY or PRINT statement. If all of them are omitted, VERIFY defaults to verifying the last track of every cylinder, and PRINT defaults to printing track 0 from each of the first 100 cylinders on each selected DASD volume:

ERASEDATA=

NONE – (PRINT only) Specifies that only Record 0 (R0) is printed from each selected track, including a count of the additional data records on the track. This greatly reduces the size of the printout when printing from a volume before it has been erased.

Default: One line is printed for each record on each selected track.

ERASENUMB=

***nnnnn* –** (VERIFY and PRINT only) For VERIFY, specifies the number of tracks (1-65535) that must fail erase verification on a DASD volume before the VERIFY function stops processing the volume. For PRINT, specifies the number of tracks (1-65535) whose contents are printed from each DASD volume. The tracks that are verified or printed are controlled by the operands ERASESTARTCYL=, ERASESTARTTRK=, and ERASESKIP=, described below. See FDR175 for details of the print format.

Default: 100.

ERASESKIP=

***nnnnn* –** (VERIFY and PRINT only) ERASESKIP=, in conjunction with ERASESTARTCYL= and ERASESTARTTRK=, described below, specify the tracks to be verified or printed. The first track is specified by ERASESTARTCYL= and ERASESTARTTRK=, and subsequent tracks are selected by adding the ERASESKIP= value (1-32767) to the relative track number of the last track printed. If you want to print contiguous tracks, specify ERASESKIP=1.

Default: 15 (verify or print one track from contiguous cylinders).

ERASESTARTCYL=

ERASESTARTTRK=

(VERIFY and PRINT only) ERASESTARTCYL= specifies the cylinder number (0-65535) of the first track to verify or print, and ERASESTARTTRK= specifies the track number (0-14) of the first track to verify or print. Both values are in decimal.

Default: VERIFY defaults to ERASESTARTCYL=0 and ERASESTARTTRK=14 which is the last track of the first cylinder. If ERASESKIP= is set to or defaults to 15, then the last track of every cylinder is verified.

Default: PRINT defaults to 0 for both operands. If both are omitted, the print starts with the label track (cylinder 0 track 0).

NOTE: To VERIFY the entire DASD volume, specify operands ERASESTARTCYL=0, ERASESTARTTRK=0, and ERASESKIP=1.

ONLINE=

BYPASS – Verify or print only offline DASD

VERIFY – (VERIFY only) Verify ONLINE DASD as well as OFFLINE DASD. Note that ONLINE DASD may fail verification if tracks in the VTOC, VTOCIX, or VVDS are verified. You can use the `ERASESTARTCYL=` and/or `ERASESTARTTRK=` operands to start the verification after the VTOC/VTOCIX/VVDS.

Default: For VERIFY, only OFFLINE DASD are verified. Any ONLINE DASD selected by the MOUNT statement are bypassed with a diagnostic message.

Default: For the PRINT command, ONLINE and OFFLINE volumes are printed.

MOUNT STATEMENT SYNTAX

The syntax of the MOUNT statement is:

```
MOUNT     ERASEUNIT=(uuu1[,uuu2,…])   ,VTOCCYL=cccc
                                        ,VTOCTRK=tt|
          ,CHANGEVOL=vvvvvv           ,VTOCLOC=nnnnn
          ,CHANGEVOLNUM=0|            ,VTOCSIZE=nnnn
                        nnnnn
          ,CHECKTARGETVOL=vvvvvv
          ,STORAGEGROUP
```

One or more MOUNT ERASEUNIT= statements follow the main statement and each specifies a DASD device or set of devices to be erased, verified or printed. Only DASD devices that are offline on the system where FDRERASE is executing are selected unless ONLINE=VARYOFF is specified on the main statement. However, if the main statement is a PRINT statement, then ERASEUNIT= can specify any offline or online DASD device.

The DASD selected by each MOUNT statement are completely processed before the next MOUNT statement is examined. MAXTASKS= (on the main statement) applies only to the DASD selected on a single MOUNT statement, so if the MOUNT specifies only one or a few DASD, only those DASD are erased concurrently. The ability to enter multiple MOUNT statements is provided primarily to allow varying values for the new VTOC and volume serial on the erased DASD. It is suggested that unless you need to vary the VTOC location and size or volume serial, you provide a single MOUNT statement and identify all of the DASD to be erased on that statement.

MOUNT STATEMENT OPERANDS

ERASEUNIT=

uuuu **–** Specifies the z/OS device addresses of the DASD devices to be erased, verified, or printed. The address can be specified as a 4-digit (hex) z/OS device address, or it can be specified as 1, 2, or 3 digits with a trailing asterisk (*); in this case all eligible offline z/OS DASD addresses starting with the prefix specified are erased. To erase multiple devices or ranges of devices (by prefix), specify them in parentheses, separated by commas, but you cannot specify more than 255 subparameters (device addresses or address prefix ranges) within the parentheses. If you use address prefixes, the total number of devices to erase can be up to 8190.

Only devices that are offline are erased (but online or offline DASD can be printed). However, if you specify ONLINE=VARYOFF, online devices that can be placed offline on this system are also erased (this is intended mainly for use at disaster/recovery sites). If a device is manually placed offline after the erase step starts, it is not selected. For a VERIFY function, only offline DASD are verified; you can specify ONLINE=VERIFY if you want to verify online DASD as well as offline DASD.

NOTE: If the range to be erased includes devices that are in your I/O configuration but which do not really exist (are not in the hardware configuration of the DASD subsystem), you may receive this console message:

```
IOS002A dev, NO PATHS AVAILABLE
```

for each such device if it is the first time that the device has been accessed since the last IPL. The messages can be ignored. They may occur only under certain releases of OS/390.

For example,

```
MOUNT ERASEUNIT=17C0 erases one DASD 17C0.
MOUNT ERASEUNIT=17C* erases DASD in the range 17C0-17CF.
MOUNT ERASEUNIT=17*  erases DASD in the range 1700-17FF.
MOUNT ERASEUNIT=(17*,18*,19A*) erases DASD in the ranges
1700-17FF, 1800-18FF, and 19A0-19AF.
```

CHANGEVOL=

Allows you to change the volume serial of the erased volume, after the erase is complete. `CPYVOLID=YES` is assumed on the main statement. The value must be six characters with these values:

***** – The equivalent character is copied from the original volume serial on the DASD.

/ – Insert a sequence number into the new serial. You can put 1 to 6 slashes in the name. See CHANGEVOLNUM=.

&*uuu* – This string is replaced with the DASD device address.

Any other character – That character is inserted in that position in the new volume serial.

For example,

```
CHANGEVOL=ABC*** changes the volser to ABC plus the
                original last 3 characters.
CHANGEVOL=**XY** puts XY in the 3rd and 4th positions,
                copying the original characters in the
                other positions.
CHANGEVOL=XX&UUU changes the volser to XX plus the device
                address (for example, XX17CA).
CHANGEVOL=ABCXYZ changes the volser to ABCXYZ.
CHANGEVOL=AB///Z changes the volser to ABnnnZ. "nnn" id
                incremented for each new volume.
```

CHANGEVOLNUM=

***nnnnn* –** Specifies the starting sequence number to be used when the `CHANGEVOL=` value contains one or more / (slashes). The value can be from 0 to 65535.

NOTE: The maximum sequence number generated is 65535.

Default: 0.

CHECKTARGETVOL=

If specified, the volume serial of the volume to be erased is compared to the value specified. The value can be a complete volume serial (up to six characters) or it may be a volume serial prefix followed by an asterisk (`*`). For example, `CHECKTARGETVOL=PAY001` or `CHECKTARGETVOL=PAY*`. If the volume has a proper volume label but the volume serial does not match, it is not erased. If it has no volume label or if it was previous erased by FDRERASE or was an FDRPAS source volume, the comparison is not done and the volume is erased. The operand can be abbreviated `CHKTVOL=`.

NOTE: `CHECKTARGETVOL=`*`vvvvvv`* (a complete volume serial) is usually appropriate only when `ERASEUNIT=`*`uuuu`* specifies only a single device. However, if there are multiple MOUNT statements in an FDRERASE job, FDRERASE processes them one at a time, so only one volume is erased at a time.

STORAGEGROUP

The erased VTOC has the "SMS-managed" flag turned on. This is honored only if the VTOC is being rebuilt (`CPYVOLID=YES` or `EMPTYVTOC` on the main statement). You must add the volume serial of the erased DASD to your SMS configuration before it can be used as an SMS-managed volume.

Default: If the DASD had a valid volume label and VTOC, and the SMS flag was on in the original VTOC, it remains on in the erased VTOC. Otherwise, it is off.

VTOCCYL=
VTOCTRK=
VTOCLOC=

Specifies the new starting location of the VTOC, if CPYVOLID=YES or EMPTYVTOC is specified on the main statement. You can specify the starting cylinder number (VTOCCYL=*cccc*) and starting track (VTOCTRK=*tt*). If VTOCCYL= is specified but not VTOCTRK=, the starting track defaults to 0. VTOCTRK= cannot be specified without VTOCCYL=.

Alternately, you can specify the starting track of the VTOC, relative to track 0 on the volume (VTOCLOC=*nnnnn*).

If you specify one of these operands to specify the start of the new VTOC, you must also specify the size of the VTOC (VTOCSIZE=).

Because of IBM requirements, the last track of the VTOC can be no higher than relative track 65535 (cylinder 4368), so the starting track of the new VTOC plus the new VTOC size cannot exceed 65535.

Default: The VTOC (and VTOCIX if present) is written in its original location with its original size.

VTOCSIZE=

***nnnn* –** Specifies the new size of the VTOC, in tracks (1-9999). VTOCSIZE= must be specified in conjunction with parameters specifying the starting track of the new VTOC (see above).

VTOC NOTES

If the main statement specified the EMPTYVTOC operation or the CPYVOLID=YES operand, then ICKDSF is invoked at the end of the operation to build a VTOC (and possibly a VTOCIX).

The operands above (VTOC*xxxx*=) can be used to specify the location and size of the empty VTOC. If they are not specified and if the volume already has a VTOC, the empty VTOC has the same location and size. If there was an indexed VTOC (VTOCIX) on the volume, even if it was not active, it is rebuilt and activated with the same location and size.

For an EMPTYVTOC operation, if the volume did not previously have a valid volume label and VTOC, then the VTOC size and location must be specified or the EMPTYVTOC fails. A new VTOCIX is created, $^1/_{16}$th the size of the VTOC (rounded up, with a minimum of three tracks) and is placed immediately after the new VTOC. This provides a very quick and easy way to initialize a set of volumes with a VTOC and VTOCIX.

If you specify the new VTOC size and location, and the volume originally contained a VTOCIX or did not contain a VTOC, a new VTOCIX is created, $^1/_{16}$th the size of the VTOC (rounded up, with a minimum of three tracks) and is placed immediately after the new VTOC.

EXCLUDE STATEMENT SYNTAX

The syntax of the EXCLUDE statement is:

EXCLUDE	ERASEUNIT=*uuuu*

An EXCLUDE statement identifies a volume or a group of volumes that are not to be processed. One or more EXCLUDE statements may be present and must follow the main statement and precede the MOUNT statement(s).

The control statements are always scanned in the order that they were input; therefore, EXCLUDE statements must precede MOUNT statements.

Example 1. Select all units starting with “07C” except unit address “07C0”.

```
EXCLUDE  ERASEUNIT=07C0
MOUNT    ERASEUNIT=07C*
```

Example 2. Select all units starting with “07” except the units starting with “07C” and “07D” as well as unit “07E0”.

```
EXCLUDE  ERASEUNIT=07C*
EXCLUDE  ERASEUNIT=07D*
EXCLUDE  ERASEUNIT=07E0
MOUNT    ERASEUNIT=07*
```

330.5 FDRERASE CONSIDERATIONS

WHY ERASE DASD?

Your corporate data is a valuable resource. The data often includes information that you do not want to share with anyone outside your company, and sometimes not even with others in your company. Government regulations (for example, HIPAA, Sarbanes-Oxley, and others) as well as corporate standards may legally obligate you to protect the data from unauthorized access. You may have security procedures that limit access to data in your data center and control access when data must be sent out of the data center.

Yet, in many cases, little attention is paid to residual data left on DASD after data sets are deleted or moved to other locations. Deleting a data set from DASD does not erase the data from the DASD tracks, it just deletes the VTOC pointers to the tracks containing the data (there is a security system ERASE option that will actually erase deleted data sets but it is rarely used because of overhead).

When a DASD volume has been emptied of data, or is no longer needed, you may need to erase the DASD to be sure that no residual data remains on the DASD volume, so that no future user of the DASD can retrieve your data. This is especially true if the DASD will be removed from your data center and sold to a new owner or returned to the DASD vendor. You may wish to erase the DASD even if they will be reused within your data center, so that the new users cannot retrieve unauthorized residual data. Even if you plan to scrap the DASD subsystem, it may be safest to erase the data first.

At a disaster recovery center, at the end of your disaster test, and when leaving after a real disaster, you should erase the D/R DASD volumes to ensure that the next customer using the D/R DASD cannot access your data.

ERASE STANDARDS

The National Computer Security Center (NCSC), a former division of the US National Security Agency (NSA), has documented DoD (Department of Defense) guidelines for erasing computer DASD. These definitions are found in document NCSC-TG-025 *A Guide to Understanding Data Remanence in Automated Information Systems*, also called the "Forest Green Book". You can find copies of this document online by searching the Internet for "NCSC-TG-025".

The Department of Defense has also issued DoD 5220.22-M *National Industrial Security Program Operating Manual* with guidelines for erasing DASD. You can find copies of this document by searching for "5220.22-M". In addition, there is a memorandum from the Assistant Secretary of Defense (ASD C3I), dated 4 Jun 2001, on *Disposition of Unclassified DoD Computer Hard Drives*.

The Computer Security Division of the US National Institute of Standards and Technology is has published Special Publication 800-66 Rev-1: *An Introductory Resource Guide for Implementing the Health Insurance Portability and Accountability Act (HIPAA) Security Rule* that documents DASD overwriting requirements. You can read this publication on the Internet at http://csrc.nist.gov/publications/PubsSPs.html.

BSI, the German Federal Office for Information Security, has issued the *IT Baseline Protection Manual* that can be found on the Internet at www.bsi.bund.de. Section 2.167 "Secure deletion of data media" discusses requirements for overwriting DASD data.

The *Australian Government Information Security Manual* defines overwriting requirements for DASD data. This document can be found on the Internet at www.dsd.gov.au.

The Department of Defense requirements for media sanitization require an overwrite with a pattern, and then its complement, followed by another unclassified pattern (for example, "00110101" followed by "11001010" and then followed by "10010111". This series is considered three cycles. For these government requirements, sanitization is not complete until six passes of three cycles are successfully completed.

There may be other data erasure requirements in other countries or industries.

Some government security and privacy requirements require that data be kept secure from outside access but do not specify compliance techniques in detail. FDRERASE can be used as part of the process to ensure that such requirements are met.

330.5 FDRERASE CONSIDERATIONS

DATA THREAT LEVELS

There are three levels of threat of unauthorized access to your data that are addressed by the different types of erase functions performed by FDRERASE. Modern DASD subsystems use internal fixed-block architecture (FBA) DASD to store the count-key-data (CKD) data used by z/OS systems; this architecture is described in more detail later in this section:

1. The first threat level is that your count-key-data (CKD) data can be accessed by another z/OS program or another z/OS system. This threat might by a deliberate attempt to access unauthorized data. It also might be inadvertent access, for example, data left at a disaster/recovery site after a test or real disaster, accessed by a subsequent D/R customer.
2. A higher threat level is that the fixed-block architecture (FBA) DASD can be removed from the vendor's DASD subsystem, attached to another system (for example, a PC) as an FBA DASD (SCSI, Fibre, etc) and accessed by an FBA program with little special programming or hardware required. This could occur inadvertently if the FBA DASD from a decommissioned count-key-data (CKD) subsystem are removed and sold or reused, but it could also be a deliberate attempt to access your data.
3. The highest threat level is that the fixed-block architecture (FBA) DASD can be removed (as in #2 above) and special FBA access techniques or equipment are used to access previous versions of data on the DASD, even if the data has been overwritten. The write hardware of FBA DASD may rewrite data at slightly varying locations, leaving the possibility of residual magnetic signatures of the previous contents; this residual data can only be read with special hardware or programming, as used by data recovery companies. Using special hardware or programming to read the data is definitely a deliberate attempt to access your data.

TYPES OF ERASE

ERASE

The ERASE function of FDRERASE addresses threat level #1 and #2. By default, it writes a track-length record of binary zeros on every track, insuring that the data on the fixed-block architecture (FBA) DASD is completely overwritten. You can optionally request that this record be written multiple times (`ERASEPASS=`*n*) and you can specify the data byte to be used instead of zero for each such pass. If the data byte is zero (the default), then FDRERASE actually sends only a few bytes per track down the DASD channel (the control unit pads the record with additional zeros), so the default FDRERASE is quite fast. There are also options to write a random pattern, or to erase the track.

The default ERASE function meets the NCSC and DoD definition of "clearing" or "overwriting" the DASD.

The default ERASE function also meets the Australian guideline for sanitizing unclassified media.

SECUREERASE

The SECUREERASE function of FDRERASE addresses all three threat levels. SECUREERASE writes a minimum of three passes of data (track-length records) on each track with a varying pattern of data in each pass. SECUREERASE prevents even sophisticated recovery techniques from recovering data from the fixed-block architecture (FBA) DASD. Although this usually takes more time than other erase options, this can provide the highest assurance that your most sensitive data has been obliterated.

SECUREERASE meets the NCSC definition of "purging" the DASD and the DoD definition of "sanitizing" the DASD.

SECUREERASE meets the HIPAA requirements for disposal and reuse of DASD that contain protected health information.

SECUREERASE with `ERASEPASS=4` *or* `6` *meets the German BSI requirements for overwriting DASD.*

SECUREERASE meets the Australian requirements for sanitizing In-Confidence and Restricted media, and SECUREERASE with `ERASEPASS=5` *meets their requirement for sanitizing Protected media.*

SECUREERASE with `ERASEPASS=18` *meets the requirements for sanitizing media by completing six passes of three cycles of overwriting with a pattern, and then its complement, and finally with another unclassified pattern (for example, "00110101", followed by "11001010", and then followed by "10010111" (three cycles).*

VERIFY

Some erasure guidelines require that the erasure be verified by checking a percentage of the erased DASD. The VERIFY function of FDRERASE can be used to meet this requirement. VERIFY will read selected tracks from the specified volumes and verify that the tracks contain either no records or a full-track record created by FDRERASE. By default, it samples the last track in every cylinder on the volume, but you can direct it to sample more tracks, up to the entire volume.

VERIFY meets the DoD requirement for "verifying" the DASD were actually erased.

ERASING MODERN DASD

All modern DASD subsystems use internal fixed-block architecture (FBA) DASD to emulate the count-key-data (CKD) DASD that are used by z/OS systems. Every DASD vendor has a different scheme for storing the emulated CKD data onto the FBA DASD, but in most of them, there is a fixed FBA location for each emulated CKD track. FDRERASE is able to overwrite the emulated CKD tracks to make the original data unavailable.

However, there are some considerations:

- The fixed-block architecture (FBA) DASD are usually off-the-shelf DASD that can be removed from the count-key-data (CKD) DASD subsystem and attached to another system as an FBA DASD. Depending on how the DASD vendor has written your CKD data, it may be possible to recover your data directly from the FBA DASD. Running ERASE or SECUREERASE on the CKD volumes makes this difficult or impossible.
- Most modern DASD subsystems are reconfigurable, meaning that the mapping of the emulated count-key-data (CKD) DASD volumes onto the fixed-block architecture (FBA) DASD can be changed. If this mapping is changed, some of your old data may reside in areas of the FBA DASD that are no longer in use. This data may be recoverable if the FBA DASD are removed from the subsystem, so you should run FDRERASE on the CKD DASD volumes involved before reconfiguring the subsystem.
- Most modern DASD subsystems use “hot spares” for the fixed-block architecture (FBA) DASD. If an FBA DASD fails, the subsystem is able to assign an unused hot spare FBA DASD to replace the failed DASD, and recreate the data that was on the failed DASD. Usually the DASD vendor is automatically notified and the failed DASD will be promptly replaced and returned to the vendor for diagnosis and repair. However, your data may still be on that failed DASD, and may be recoverable. FDRERASE cannot access the failed FBA DASD even before it is replaced and cannot erase data on it.

The fixed-block architecture (FBA) DASD used in DASD subsystems usually have a capacity that far exceeds the capacity of the emulated count-key-data (CKD) DASD, so a single FBA DASD may contain many emulated DASD volumes. If the DASD subsystem is using RAID-5 or RAID-10 configurations, then a single emulated DASD is spread across several physical FBA DASD (usually called a “RAID rank”).

When many count-key-data (CKD) DASD that reside on the same physical fixed-block architecture (FBA) DASD or RAID rank are erased at the same time, the contention for the disk heads and data paths may severely degrade the total performance of the FDRERASE job. So, FDRERASE uses hardware queries (which vary by DASD vendor) to identify the underlying physical DASD or rank, and limits the number of active erase tasks on any one underlying DASD or rank (see “MAXEU=” in Section 330.4). FDRERASE identifies the vendor, control unit serial number, SSID (Subsystem ID), and internal DASD disk/rank ID (see “Vendor Considerations for Erasing” in Section 330.5).

If any of this information cannot be determined, FDRERASE simply uses the z/OS device address of the logical DASD as the ID.

To get the best elapsed time when erasing a large number of DASD, specify many DASD devices on a single MOUNT statement, so that FDRERASE can manage the erase tasks, spreading the activity across various subsystems and internal DASD.

VENDOR CONSIDERATIONS FOR ERASING

EMC

EMC subsystems allow FDRERASE to determine if a given DASD is online to other systems. If the volume has a valid volume label, FDRERASE makes this check and does not allow the DASD to be erased unless it is offline to all systems (you can override this check by specifying ACTIVETARGET=PROCESS but you must be sure that the volume is truly not in use). This capability is not available on subsystems from other vendors.

If you use BCVs (Business Continuance Volumes under EMC TimeFinder), you must be sure that the BCVs are also erased. If the BCVs are ESTABLISHed at the time that FDRERASE is done, they are automatically erased as well. If the BCVs are SPLIT, then you need to run FDRERASE against the split BCVs. If you try to erase a BCV unit that is ESTABLISHed, you get an I/O error (either Intervention Required or SIM (time out)).

For mirrored (RAID-1) DASD, FDRERASE identifies the internal DASD by the physical disk ID for the online DASD and its mirrors. For RAID-10, the RAID group ID is used.

HDS

If you use HDS ShadowImage or FlashCopy to create duplicate volumes, you must be sure that the duplicate volumes are also erased. If the ShadowImage volumes are ESTABLISHed at the time that FDRERASE is done, they are automatically erased as well. If the volumes are SPLIT or SUSPENDed or created with FlashCopy, then you need to run FDRERASE against the duplicate volumes as well. If you try to erase a ShadowImage unit that is ESTABLISHed, you get an I/O error (either Intervention Required or SIM (time out)).

FDRERASE identifies the internal DASD by the RAID ID.

IBM

If you have used FlashCopy to create duplicate volumes, you must erase the FlashCopy target volumes as well. You should terminate any active FlashCopy sessions involving the DASD to be erased; otherwise, the elapsed time of the erase increases substantially.

FDRERASE identifies the internal DASD by the RAID rank ID.

RVA/SVA

DASD subsystems with a “virtual” architecture, such as the IBM RVA and Oracle StorageTek SVA are difficult to erase since the subsystem writes each updated track in a new location on the fixed-block architecture (FBA) DASD. The original data on each track is not overwritten although the virtual pointers to the data tracks no longer point to the original data. FDRERASE will make all the data in a virtual DASD subsystem unavailable, but it may still be possible to recover some of it if the FBA DASD are removed from the subsystem. Data in these virtual subsystems is compressed, so it will be difficult to reconstruct your data from the FBA DASD, but if you must be sure that it is physically erased or overwritten, consult the DASD vendor.

If you are planning to remove the RVA/SVA from your data center, INNOVATION DATA PROCESSING recommends the following to ensure that the data on the fixed-block architecture (FBA) DASD is overwritten and not recoverable:

- Run an ERASE with `ERASEPATTERN=FE` on all volumes in the subsystem. The erase patterns created by `ERASEPATTERN=FE` are minimally compressible by the RVA/SVA compress algorithm, so the data tracks are written to the back-end fixed-block architecture (FBA) DASD with be close to the original lengths. Do not use the `VARYON=AFTER` option so that the DASD are left offline, preventing the IXFP/SVAA software from releasing the erased tracks. Eventually, the Net Capacity Load (NCL) exceeds the capacity of the back-end DASD (generating I/O error messages) that ensure that all of your original data has been overwritten; at that point you can cancel the ERASE job.

Because of the virtual architecture of the RVA/SVA, FDRERASE cannot identify internal disk IDs for these DASD subsystems.

PPRC/HRC/SRDF/Global Mirror for z/Series (XRC)

There are special considerations for volumes that have remote copies created by an active PPRC, Global Mirror for z/Series (XRC), HRC (Hitachi), or SRDF (EMC) session:

By default, FDRERASE does not erase volumes that have a valid volume label and are in an active PPRC, HRC, or SRDF session; they are bypassed with an explanatory message. An active remote copy may indicate that the data on the volume is still needed, so FDRERASE defaults to protecting those volumes. FDRERASE does not currently test for Global Mirror for z/Series (XRC).

If you do want to erase this DASD, you can terminate the remote session and rerun the FDRERASE job.

If you also want to erase the remote copies of the DASD, there are two alternatives:

- If the remote copy session is still active at the time of the erase, you can specify `ACTIVETARGET=PROCESS` to allow the erase to proceed; both the local and remote DASD are erased. However, this may greatly slow down the erase process since each erase or overwriting track must be sent to the remote control unit.
- If the remote DASD also have a channel connection to the local processor or a remote processor, it is usually faster to terminate the remote sessions and erase the remote volumes directly with FDRERASE.

ALTERNATE TRACKS

Older DASD subsystems that use “real” count-key-data (CKD) DASD (instead of emulating the CKD DASD on internal fixed-block architecture (FBA) DASD) included a set of “alternate tracks” on every volume. These alternate tracks were used when a track on the DASD was found to be defective; an available alternate track was assigned in place of the defective primary track. The process of assigning an alternate track automatically erased any data on the original defective primary track; conversely, if a primary track is reclaimed, data is erased from the alternate track that used to be assigned.

When FDRERASE is used to erase a real count-key-data (CKD) DASD, it automatically accesses any assigned alternate tracks and erases them. Alternate tracks that are not currently assigned are not erased, but they will contain no customer data.

On all modern DASD subsystems that use internal FBA DASD there are no CKD alternate tracks. On some vendor's subsystems, the logical CKD DASD may appear to have a set of alternate tracks, but they cannot be assigned so they are never used for customer data.

It is possible that defective sectors on the FBA DASD in a DASD subsystem have been assigned to alternate sectors, but this is under control of the DASD control unit and cannot be seen or affected by FDRERASE, so defective FBA sectors will never be erased by FDRERASE, and they may contain some fragmentary residual customer data.

VERIFICATION AND AUDITING

The standards, regulations, or guidelines that require that the data be erased may also require that you verify that the data was truly erased; this verification may need to be done by another person. The VERIFY and PRINT functions of FDRERASE can be used to perform this verification. Depending on the requirements, it may be adequate to verify only a subset of the erased DASD and only a subset of the tracks on those DASD volumes. By default, VERIFY verifies one track in every cylinder on the selected volumes; this is 1/15th (6.7% of each volume).

Those standards, regulations, or guidelines may also require that records be kept of the DASD erased for a period of time, even after the DASD have been reused or removed, for auditing purposes. You should save the complete FDRERASE job outputs, including the job log (system messages), SYSPRINT (SWAP task messages), SYSPRT*xx* (subtask messages), and FDRSUMM (one-line summaries) DD statement output.

FDRSUMM contains sufficient detail of the DASD erased, including the type of ERASE function, device numbers, control unit manufacturer and serial number, volume size, and tracks erased or verified. The FDRSUMM output, by itself, may be adequate for record keeping purposes.

FDRPAS USERS

With the default of `CHECKTARGET=YES`, FDRERASE erases DASD devices that were the source volumes of a successful swap, or the target devices of a successful SWAPDUMP. Do not specify `CHECKTARGET=NO` if only FDRPAS DASD volumes are to be erased.

FDRERASE and FDRPAS both check to be sure that the DASD is not already being used by the other program, so you cannot accidentally erase a DASD that is being swapped, or swap to a DASD that is being erased.

If you plan to erase DASD from a subsystem while swapping other volumes in the same subsystem, be aware that the FDRERASE I/Os may increase the FDRPAS swap elapsed time. INNOVATION DATA PROCESSING recommends that you only erase a few volumes at a time (`MAXTASKS=`*n*) in this case.

If the old DASD subsystem will be sold or returned to the vendor, you should run an ERASE or SECUREERASE to ensure that your corporate data is gone

WARNING: ***When swapping volumes with FDRPAS, the source DASD serve as a backup in case of problems with the new hardware. Do not begin erasing data from the old DASD until you are sure that the new ones are operating without problems.***

IBM TDMF PRODUCT

Users of IBM TDMF need an extra step before the old source DASD from a migration can be erased with FDRERASE. IBM TDMF sets the old source volumes offline but leaves a volume serial filled in the UCB. Validity checks within FDRERASE bypass these DASD volumes because of the volume serial. z/OS allows such a DASD volume to be allocated and used by a job, so FDRERASE cannot be certain that the DASD volume is not in use.

To clear the volume serial from the UCB, you must vary the DASD online to z/OS and then back offline with console commands such as:

```
VARY (3C00-3CFF),ONLINE
VARY (3C00-3CFF),OFFLINE
```

After this operation, FDRERASE selects and erases the DASD. An IPL also clears the volume serials.

You must also specify `CHECKTARGET=NO` when erasing the source DASD since IBM TDMF leaves a valid VTOC on the DASD with all of the original data sets.

SECURING FDRERASE

Since erasing DASD obviously offers the capability of accidentally destroying a great deal of valid data, you want to control who can execute FDRERASE.

If FDRERASE has been installed in a separate load library, you can use your security system to limit access to that library and use STEPLIB to access the library. This method is the simplest and most secure way of limiting the use of FDRERASE.

Whenever FDRERASE is executed, it checks for authority to this security resource:

```
CLASS=FACILITY    ENTITY=FDRERASE.ERASE
```

If you have defined that FACILITY class resource, then only users who have at least READ authority to the resource are able to execute FDRERASE. If you have not defined the resource, then all users are able to erase DASD.

If the job specifies `CHECKTARGET=NO`, which allows any offline DASD to be erased, FDRERASE additionally checks for READ authority to:

```
CLASS=FACILITY    ENTITY=FDRERASE.ERASEALL
```

If a user is authorized to `FDRERASE.ERASE` but not to `FDRERASE.ERASEALL`, then they are able to erase empty and FDRPAS DASD, but not other offline DASD (jobs that specify `CHECKTARGET=NO` receive a control statement error).

If the job specifies `ONLINE=VARYOFF`, which allows online DASD to be varied offline and erased, FDRERASE additionally checks for READ authority to:

```
CLASS=FACILITY    ENTITY=FDRERASE.ONLINE.VARYOFF
```

If the user is not authorized and `ONLINE=VARYOFF` is specified, the job fails with a control statement error.

If any of these FACILITY resources are not protected in your security system, then the FDRERASE functions are allowed.

CONSOLE COMMANDS FOR FDRERASE

While an FDRERASE job is running, you can stop it prematurely (before all DASD are erased) or modify the number of ERASE tasks (`MAXTASKS=`) dynamically with console commands. You can also display the status of running erase tasks.

To stop FDRERASE, issue the `STOP (P)` command with the name of the FDRERASE job:

```
P jobname
```

FDRERASE terminates as soon as all DASD currently being erased have terminated; no new DASD is started.

To dynamically modify the value of `MAXTASKS=` (the number of concurrent ERASE tasks), issue the `MODIFY (F)` command:

```
F jobname,MAXTASKS=nn
```

If you are increasing the current value of `MAXTASKS=` (to a maximum of 64), FDRERASE starts new ERASE subtasks if it has DASD that have not yet started. If you are decreasing `MAXTASKS=`, FDRERASE eventually reduces the number of active ERASE tasks to the new value; this may take a few minutes.

To display the status of active erase tasks, issue the `MODIFY (F)` command

```
F jobname,STATUS  (or just STA)
```

This responds with message on the console and in the job log of the FDRERASE job in a format similar to that displayed by the FDRERASE ISPF displays, for example,

F *JOBNAME*,STATUS

```
FDRW08  UNIT  %  PASS STATUS       MM:SS  CYL/SEC SEC/CYL ERASED  CYLS
FDRW08  41C4  86  1/1 ACTIVE ERASE  2:55      10    0.092   1902  2226
FDRW08  21C0  96  1/1 ACTIVE QUICK  2:56      54    0.018   9549 10017
```

330.6 FDRERASE EXAMPLES

All examples in this section are found in the JCL library installed with FDRERASE. The member names are PA33006*x*.

SIMERASE EXAMPLE

SIMERASE validates your control statements, and it lists the offline DASD volumes that will be erased once SIMERASE is changed to one of the other options.

```
//SIMERASE EXEC PGM=FDRERASE,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrerase.loadlib
//SYSPRINT DD SYSOUT=*
//FDRSUMM  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  SIMERASE TYPE=FULL
  MOUNT    ERASEUNIT=(3A*,3B*,4*)
/*
```

ERASE A RANGE OF FDRPAS SOURCE DASD VOLUMES EXAMPLE

After swapping a set of volumes from an old DASD subsystem to a new subsystem with FDRPAS, and verifying that the new subsystem is operating correctly, the old subsystem is to be disconnected. FDRERASE with the default ERASE function is used to clear (overwrite) all corporate data on the DASD in the old subsystem, device addresses 07C0 through 07FF. Only offline DASD in the range is erased (FDRPAS leaves them offline). The default of `CHECKTARGET=YES` verifies that each DASD device is an FDRPAS source DASD (or an empty DASD) before erasing it.

WARNING: ***When swapping volumes with FDRPAS, the source DASD serve as a backup in case of problems with the new hardware. Do not begin erasing data from the old DASD until you are sure that the new ones are operating without problems.***

```
//ERASE    EXEC PGM=FDRERASE,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrerase.loadlib
//SYSPRINT DD SYSOUT=*
//FDRSUMM  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  ERASE    TYPE=FULL
  MOUNT    ERASEUNIT=(07C*,07D*,07E*,07F*)
/*
```

SECUREERASE SELECTED FDRPAS SOURCE DASD EXAMPLE

Certain DASD in the old subsystem used to contain extremely sensitive data. FDRERASE with the SECUREERASE function is used to purge all corporate data from those DASD volumes. Each track is overwritten three times with random patterns (described earlier). All these DASD volumes must be offline (that is how FDRPAS leaves them). The default of `CHECKTARGET=YES` verifies that each DASD device is an FDRPAS source DASD (or an empty DASD) before erasing it. `ALREADYERASED=PROCESS` is specified in case some of the DASD were previously erased.

```
//SECERASE EXEC PGM=FDRERASE,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrerase.loadlib
//SYSPRINT DD SYSOUT=*
//FDRSUMM  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  SECUREERASE TYPE=FULL,ALREADYERASED=PROCESS
  MOUNT    ERASEUNIT=(07C5,07D2,07D3,07DA,
                      07E5,07F4)
/*
```

ERASE WITH PASSES AND PATTERNS EXAMPLE

A certain DASD volume used to contain extremely sensitive data. FDRERASE with the ERASE function is used to purge all corporate data from the DASD, specifying five passes (number of overwrites) for each track and the patterns to be used for each pass. All these DASD volumes must be offline (which is how FDRPAS leaves them). The default of CHECKTARGET=YES verifies that each DASD device is an FDRPAS source DASD (or an empty DASD) before erasing it. This type of ERASE may take substantially longer than other options because each track must be overwritten multiple times. The next to last pattern of “FE” generates a random pattern. The last pattern byte of “01” causes the tracks to be hardware erased in the last pass, leaving no records on them.

```
//ERASE    EXEC PGM=FDRERASE,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrerase.loadlib
//SYSPRINT DD SYSOUT=*
//FDRSUMM  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  ERASE    TYPE=FULL,ERASEPASS=5,ERASEPATTERN=AA5533FE01
  MOUNT    ERASEUNIT=73F2
/*
```

ERASE DISASTER SITE DASD EXAMPLE

When leaving a disaster/recovery site, you want to erase all corporate data that you restored. Any volumes in the selected ranges are varied offline if necessary. All the selected DASD is erased except the volumes with unit address “3020” and “3021” that containing the OS.

```
//ERASE    EXEC PGM=FDRERASE,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrerase.loadlib
//SYSPRINT DD SYSOUT=*
//FDRSUMM  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  ERASE    TYPE=FULL,CHECKTARGET=NO,MAXTASKS=64,ONLINE=VARYOFF
  EXCLUDE  ERASEUNIT=3020
  EXCLUDE  ERASEUNIT=3021
  MOUNT    ERASEUNIT=(3*,4*,5*)
/*
```

ERASE AND RELABEL DISASTER SITE DASD EXAMPLE

When leaving a disaster/recovery site, you want to erase all corporate data that you restored. In the first step, any volumes in the selected ranges are varied offline (if necessary) and erased.

The disaster site may require you to relabel and vary online the DASD that you used, so the second step can be used to do so. EMPTYVTOC builds an empty VTOC and VTOCIX on each volume. The volumes are re-labeled with SG plus the 4-digit hex unit address (for example, SG323A). An empty VTOC is built starting on relative track 1 (cylinder 0 track 1) for 44 tracks (3 cylinders). The VTOCIX is built after the VTOC (cylinder 3 track 0 for 3 tracks). Each volume is varied online and mounted after the EMPTYVTOC completes.

```
//ERASE    EXEC PGM=FDRERASE,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrerase.loadlib
//SYSPRINT DD SYSOUT=*
//FDRSUMM  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  ERASE    TYPE=FULL,CHECKTARGET=NO,MAXTASKS=64,ONLINE=VARYOFF
  MOUNT    ERASEUNIT=(3*,4*,5*)
/*
//RELABEL  EXEC PGM=FDRERASE,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrerase.loadlib
//SYSPRINT DD SYSOUT=*
//FDRSUMM  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  EMPTYVTOC   TYPE=FULL,VARYON=AFTER
  MOUNT    ERASEUNIT=(3*,4*,5*),CHANGEVOL=SG&UUU,
           VTOCLOC=1,VTOCSIZE=44
/*
```

ERASE DASD TO BE RE-USED EXAMPLE

Unused DASD, such as FDRPAS source DASD, are to be re-used for other purposes in your corporation. FDRERASE with the ERASE function is used to ensure that the previous contents of the DASD is not available to the new users of the DASD. CHECKTARGET=NO is specified because some of the DASD volumes to be erased were moved with means other than FDRPAS and may still contain old data sets to be erased. All offline DASD in the range of 3A00-3AFF are erased, up to 16 at a time. CONFERASE=YES lists all of the selected offline DASD volumes on the operator's console and requires a reply before beginning the erase. An empty VTOC and VTOCIX are rebuilt after the erase, the volumes are relabeled with TST plus the last three characters of the original volume serial (for example, LIB123 becomes TST123), and they are varied online.

WARNING: ***If these are not FDRPAS source DASD, it is your responsibility to ensure that the DASD are offline on every system in your complex and do not contain any data that must be retained.***

```
//ERASE    EXEC PGM=FDRERASE,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrerase.loadlib
//SYSPRINT DD SYSOUT=*
//FDRSUMM  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  ERASE    TYPE=FULL,CHECKTARGET=NO,MAXTASKS=16,
           CONFERASE=YES,CPYVOLID=YES,VARYON=AFTER
  MOUNT    ERASEUNIT=3A*,CHANGEVOL=TST***
/*
```

ERASE ONLINE DASD EXAMPLE

Certain DASD volumes are no longer needed and will be reused for new purposes, but you want to be sure that all previous data has been erased. FDRERASE with the ERASE function is used to overwrite the old data. ONLINE=VARYOFF is specified since the volumes are still online; FDRERASE varies them offline before erasing. The volumes must be empty, with no data sets other than the VTOC, VTOCIX, VVDS, and/or ABR Model DSCB, so you must delete all data sets before running the ERASE; if you want to erase volumes that contain data sets, specify CHECKTARGET=NO (note that this does not uncatalog any data sets).

```
//ERASE    EXEC PGM=FDRERASE,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrerase.loadlib
//SYSPRINT DD SYSOUT=*
//FDRSUMM  DD SYSOUT=*
//SYSIN    DD *
  ERASE    TYPE=FULL,ONLINE=VARYOFF
  MOUNT    ERASEUNIT=3A*
/*
```

PRINT EXAMPLE

PRINT prints information about the data on tracks from the selected DASD, which may be offline or online. This can be used to validate the information on a DASD volume before erasing it, and to verify that the data is gone after the erase. ERASENUMB=10 causes 10 tracks to be printed. By default, the print starts on cylinder 0 track 0 (the label track) and prints track 0 of each subsequent cylinder. See Section 330.7 "FDRERASE Sample Outputs" for an example of the print format.

```
//PRINT    EXEC PGM=FDRERASE,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrerase.loadlib
//SYSPRINT DD SYSOUT=*
//FDRSUMM  DD SYSOUT=*
//SYSIN    DD *
  PRINT    TYPE=FULL,ERASENUMB=10
  MOUNT    ERASEUNIT=(90C1,90C2)
/*
```

VERIFY ERASED DASD EXAMPLE

The VERIFY function of FDRERASE can be used to verify that selected DASD volumes were successfully erased. This job verifies all offline DASD in the range specified (3A00-3AFF). The VERIFY function reads the last track of every cylinder on each volume and verifies that it was erased by FDRERASE.

```
//VERIFY   EXEC PGM=FDRERASE,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrerase.loadlib
//SYSPRINT DD SYSOUT=*
//FDRSUMM  DD SYSOUT=*
//SYSIN    DD *
  VERIFY   TYPE=FULL
  MOUNT    ERASEUNIT=3A*
/*
```

NOTE: If you add the operand ONLINE=VERIFY to the VERIFY statement, FDRERASE verifies online DASD in the range specified as well as offline DASD. This may be useful to verify that all DASD in a control unit or range have been successfully erased, even if they are currently online.

COMPLETELY VERIFY A SAMPLE OF ERASED DASD EXAMPLE

This job verifies selected offline DASD. The VERIFY function reads every track of every cylinder on each volume and verifies that it was erased by FDRERASE.

```
//VERIFY   EXEC PGM=FDRERASE,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrerase.loadlib
//SYSPRINT DD SYSOUT=*
//FDRSUMM  DD SYSOUT=*
//SYSIN    DD *
  VERIFY   TYPE=FULL,ERASESTARTCYL=0,ERASESTARTTRK=0,ERASESKIP=1
  MOUNT    ERASEUNIT=(3A12,3A3C,3A40,3A44,3A52,3A75,3A82)
/*
```

EMPTY VTOCS ON FDRPAS SOURCE VOLUMES EXAMPLE

Certain FDRPAS source DASD will be reused for new purposes, but you do not need to erase the data. FDRERASE with the EMPTYVTOC function gives you a convenient way of emptying the VTOC on these DASD volumes, as an alternative to initializing them one at a time with ICKDSF. Note that the EMPTYVTOC function does not uncatalog any data sets that were in the VTOCs. CHANGEVOL= relabels the volumes and VARYON=AFTER causes them to be remounted after emptying the VTOC.

```
//EMPTYVTC EXEC PGM=FDRERASE,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrerase.loadlib
//SYSPRINT DD SYSOUT=*
//FDRSUMM  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  EMPTYVTOC   TYPE=FULL,VARYON=AFTER
  MOUNT    ERASEUNIT=21C*,CHANGEVOL=XX****
/*
```

SANITIZE DASD TO COMPLY WITH USA DEPARTMENT OF DEFENSE (DOD) STANDARDS EXAMPLE

Overwrite the DASD with six passes of three cycles (for a total of 18 passes) with a pattern, then its complement, and finally with another pattern (for example, “00110101”, followed by “11001010”, and then followed by “10010111”.)

```
//SECERASE EXEC PGM=FDRERASE,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=fdrerase.loadlib
//SYSPRINT DD SYSOUT=*
//FDRSUMM  DD SYSOUT=*
//SYSIN    DD *
  SECUREERASE TYPE=FULL,ERASEPASS=18
  MOUNT    ERASEUNIT=23C*
/*
```

PRODUCE A REPORT OF ANY VOLUMES NOT ERASED IN AN SSID EXAMPLE

After the FDRERASE job has completed, you want to ensure that all volumes in SSIDs 3000 and 3001 have been erased. Volumes may have been bypassed if they were online or the UCB range specified missed volumes.

This report is produced by FDREPORT, which is a cost option in the FDR product line. If you have an FDRERASE trial or a 90-day rental of FDRERASE, FDREPORT is included in the installation package and can be run. If you do not have FDREPORT on the installation package, please contact INNOVATION DATA PROCESSING for pricing.

```
//REPORT   EXEC PGM=FDREPORT,REGION=0M
//SYSPRINT DD SYSOUT=*
//ABRMAP   DD SYSOUT=*
//SYSIN    DD *
  DEFAULT  ENABLE=OFFLINE,SUMDEVICE=UNIQUE,TITLE=LEFT
  TITLE    LINE='UCB LIST FOR SSID <VLSSID>'
  XSELECT  VLVOLID.NE.FDR5,VLSSID=(3000,3001)
  REPORT   FIELD=(VLUNIT,VLVOLSER,VLDEVTYP,UCBSTATS,VLVOLID)
  SORT     FIELD=(VLSSID,VLUNIT),BREAK=(YES,NO)
  PRINT    DATATYPE=VOLDATA
/*
```

The output produced by this would resemble the following showing that three volumes in SSID 3001 with UCB addresses of 300D, 3012, and 3014 do not have a VOLID of FDR5, which would be on all volumes erased by FDRERASE,

```
UCB LIST FOR SSID 3001

UADR VOLSER DEVTYPE UCBSTAT VOID
---- ------ ------- ------- ----
300D SY300D 3390-3  ONLINE  VOL1
3012 SY3012 3390-54 ONLINE  VOL1
3014 SY3013 3390-27 OFFLINE FDR3
```

330.7 FDRERASE SAMPLE OUTPUTS

SIMERASE SAMPLE OUTPUT

The SIMERASE examines volumes in the range specified (07C0-07CF). SIMERASE identifies those that will be bypassed because they are online and those that are offline and will be erased. SIMERASE also identifies those that have previously been erased by FDRERASE; since ALREADYERASED=PROCESS was specified, the OVERRIDDEN indicates that they will be re-erased. One DASD contained data sets and the default of CHECKTARGET=YES causes it to be bypassed.

```
FDR001  FDR ERASE VOLUMES         - FDRERASE               - INNOVATION DATA PROCESSING                        PAGE   1
FDR303  CARD IMAGE --  SIMERASE TYPE=FULL,ALREADYERASED=PROCESS
FDR303  CARD IMAGE --  MOUNT ERASEUNIT=07C*
FDR234   ERASE BYPASSED VOL=RVA7CF - UNIT=07CF REASON=8 - TARGET ERASEUNIT IS NOT OFFLINE OR NOT CONNECTED OR HAS VOLSER
FDR234   ERASE BYPASSED VOL=RVA7CE - UNIT=07CE REASON=8 - TARGET ERASEUNIT IS NOT OFFLINE OR NOT CONNECTED OR HAS VOLSER
FDR234   ERASE BYPASSED VOL=RVA7CD - UNIT=07CD REASON=8 - TARGET ERASEUNIT IS NOT OFFLINE OR NOT CONNECTED OR HAS VOLSER
FDR234   ERASE BYPASSED VOL=RVA7CC - UNIT=07CC REASON=8 - TARGET ERASEUNIT IS NOT OFFLINE OR NOT CONNECTED OR HAS VOLSER
FDR234   ERASE BYPASSED VOL=RVA7C9 - UNIT=07C9 REASON=8 - TARGET ERASEUNIT IS NOT OFFLINE OR NOT CONNECTED OR HAS VOLSER
FDR234   ERASE BYPASSED VOL=RVA7C8 - UNIT=07C8 REASON=8 - TARGET ERASEUNIT IS NOT OFFLINE OR NOT CONNECTED OR HAS VOLSER
FDR234   ERASE BYPASSED VOL=RVA7C7 - UNIT=07C7 REASON=8 - TARGET ERASEUNIT IS NOT OFFLINE OR NOT CONNECTED OR HAS VOLSER
FDR234   ERASE BYPASSED VOL=RVA7C6 - UNIT=07C6 REASON=8 - TARGET ERASEUNIT IS NOT OFFLINE OR NOT CONNECTED OR HAS VOLSER
FDR234   ERASE BYPASSED VOL=RVA7C3 - UNIT=07C3 REASON=8 - TARGET ERASEUNIT IS NOT OFFLINE OR NOT CONNECTED OR HAS VOLSER
FDR234   ERASE BYPASSED VOL=RVA7C2 - UNIT=07C2 REASON=8 - TARGET ERASEUNIT IS NOT OFFLINE OR NOT CONNECTED OR HAS VOLSER
FDR234   ERASE BYPASSED VOL=RVA7C0 - UNIT=07C0 REASON=8 - TARGET ERASEUNIT IS NOT OFFLINE OR NOT CONNECTED OR HAS VOLSER
FDR235  FDRERASE WILL ERASE THE FOLLOWING     5 UNITS:
FDR235   07C1 07C4 07C5 07CA 07CB
FDR270   DEVICE IS ALREADY ERASED UNIT=07C1 VOL=RVA7C1 OVERRIDDEN
FDR270   DEVICE IS ELIGIBLE FOR ERASE UNIT=07C1 VOL=RVA7C1 VOLID=FDR5 CU=STK98765/0501-07C10000 00000000
FDR270   DEVICE IS ELIGIBLE FOR ERASE UNIT=07C4 VOL=RVA7C4 VOLID=VOL1 CU=STK98765/0501-07C40000 00000000
FDR270   DEVICE IS ELIGIBLE FOR ERASE UNIT=07C5 VOL=RVA7C5 VOLID=VOL1 CU=STK98765/0501-07C50000 00000000
FDR270   DEVICE IS ALREADY ERASED UNIT=07CA VOL=RVA7CA OVERRIDDEN
FDR270   DEVICE IS ELIGIBLE FOR ERASE UNIT=07CA VOL=RVA7CA VOLID=FDR5 CU=STK98765/0501-07CA0000 00000000
FDR255   ERASE TARGET UNIT 21CC CONTAINS DATA SETS FIRST DSN=ICF1.ERASE008.KSDS.DATA
FDR270   DEVICE IS NOT ELIGIBLE FOR ERASE UNIT=07CB VOL=RVA7CB VOLID=VOL1
FDR998** FDR COMPLETED WITH ERRORS
```

The following samples were generated on a test system, erasing a 3390-3 DASD volume in an IBM 2105-F20 Shark on FICON channels. The elapsed times are representative, but your elapsed times may vary depending on your system configuration, type of subsystem, and the number of concurrent erases you execute. By default, one FDRERASE step processes up to 64 DASD volumes concurrently, which may elongate the times of individual DASD but reduces the total elapsed time for all DASD specified.

ERASE SAMPLE OUTPUT

The ERASE of a 3390-3 volume took 2.2 minutes, overwriting each track one time. The defaults of one pass with a data record of all zeros were used.

```
FDR001  FDR ERASE VOLUMES         - FDRERASE               - INNOVATION DATA PROCESSING
FDR303  CARD IMAGE --   ERASE  TYPE=FULL,ALREADYERASED=PROCESS
FDR303  CARD IMAGE --   MOUNT  ERASEUNIT=(21C3)
FDR235  FDRERASE WILL ERASE THE FOLLOWING     1 UNITS:
FDR235   21C3
FDR170   DEVICE IS ALREADY ERASED UNIT=21C3 VOL=SH20E3 OVERRIDDEN
FDR170   DEVICE IS ELIGIBLE FOR ERASE UNIT=21C3 VOL=SH20E3 VOLID=FDR5 CU=IBM12345/9876-01000000 00000000
FDR172   ERASE STARTED PASS 1 PATTERN=00
FDR172   ERASE   ENDED PASS 1
FDR173   ERASE HARDENED DATA TO UNIT=21C3 IN   1 SECS             0000-0D0A-00000124
FDR241  FDRERASE SUCCESSFULLY COMPLETED ERASE OF VOL=SH20E3 ON UNIT=21C3
FDR122  OPERATION STATISTICS FOR 3390 VOLUME.................  21C3
FDR122                        CYLINDERS ON VOLUME.............3,339
FDR122                        DATASETS PROCESSED..................0
FDR122                        BYTES READ FROM DASD................0
FDR122                        DASD TRACKS ERASED.............50,085
FDR122                        NUMBER OF ERASE PASSES..............1
FDR122                        DASD EXCPS.....................3,348
FDR122                        TARGET DASD EXCPS...................0
FDR122                        CPU TIME (SECONDS)..............0.297
FDR122                        ELAPSED TIME (MINUTES).............2.2
FDR122                        ERASE TIME.........................2.2
FDR999   FDR SUCCESSFULLY COMPLETED
```

SECUREERASE SAMPLE OUTPUT

The SECUREERASE of a 3390-3 volume took 7.1 minutes, overwriting each track three times. The FDR172 messages show the three passes and the data pattern used for each one. Because of the three passes, the DASD TRACKS ERASED in the FDR122 message is three times the number of tracks on the volume.

```
FDR001  FDR ERASE VOLUMES         - FDRERASE                - INNOVATION DATA PROCESSING
FDR303  CARD IMAGE --   SECUREERASE  TYPE=FULL,ALREADYERASED=PROCESS
FDR303  CARD IMAGE --   MOUNT  ERASEUNIT=(21C3)
FDR235  FDRERASE WILL ERASE THE FOLLOWING     1 UNITS:
FDR235   21C3
FDR170   DEVICE IS ALREADY ERASED UNIT=21C3 VOL=SH20E3 OVERRIDDEN
FDR170   DEVICE IS ELIGIBLE FOR ERASE UNIT=21C3 VOL=SH20E3 VOLID=FDR5 CU=IBM12345/9876-01000000 00000000
FDR172   ERASE STARTED PASS 1 PATTERN=8F
FDR172   ERASE   ENDED PASS 1
FDR173   ERASE HARDENED DATA TO UNIT=21C3 IN   0 SECS             0685-0D0A-00000186
FDR172   ERASE STARTED PASS 2 PATTERN=70
FDR173   ERASE HARDENED DATA TO UNIT=21C3 IN   1 SECS             0000-0685-00006086
FDR172   ERASE   ENDED PASS 2
FDR173   ERASE HARDENED DATA TO UNIT=21C3 IN   0 SECS             0685-0D0A-00006086
FDR172   ERASE STARTED PASS 3 PATTERN=F9
FDR173   ERASE HARDENED DATA TO UNIT=21C3 IN   0 SECS             0000-0685-0000605B
FDR172   ERASE   ENDED PASS 3
FDR173   ERASE HARDENED DATA TO UNIT=21C3 IN   1 SECS             0000-0D0A-00006008
FDR241  FDRERASE SUCCESSFULLY COMPLETED ERASE OF VOL=SH20E3 ON UNIT=21C3
FDR122  OPERATION STATISTICS FOR 3390 VOLUME.................  21C3
FDR122                        CYLINDERS ON VOLUME.............3,339
FDR122                        DATASETS PROCESSED..................0
FDR122                        BYTES READ FROM DASD................0
FDR122                        DASD TRACKS ERASED.............150,255
FDR122                        NUMBER OF ERASE PASSES..............3
FDR122                        DASD EXCPS.....................10,038
FDR122                        TARGET DASD EXCPS...................0
FDR122                        CPU TIME (SECONDS)..............3.635
FDR122                        ELAPSED TIME (MINUTES).............7.1
FDR122                        ERASE TIME.........................7.1
FDR999   FDR SUCCESSFULLY COMPLETED
```

FDRSUMM SAMPLE OUTPUT

If any FDRERASE step (except for a SIMERASE) includes an FDRSUMM DD statement, this contains a 1-line summary of every DASD processed. The title line shows the type of FDRERASE function executed, in this example, it was an ERASE. The erase pattern(s) used or found by verification is shown on the right. If this was for a VERIFY function, the "TRACKS VERIFIED" column shows the number of tracks that were read for verification, even if they did not pass erase verification.

```
FDR001  FDR        ERASE VOLUMES  - FDRERASE                    - INNOVATION DATA PROCESSING
        COMP ELAPSED         VOLUME                        TRACKS   # OF     TRACKS       ERASE
DEVICE  CODE TIME(MIN)     SIZE(CYL)      SERIAL #       VERIFIED PASSES     ERASED  PATTERN BYTE(S)
  07C1     0      0.1         3,339      STK 10030              0      1     50,085  00
  07C4     0      0.1         2,226      STK 10030              0      1     33,390  00
  07C5     0      0.1         2,226      STK 10030              0      1     33,390  00
  07CA     0      0.1         1,113      STK 10030              0      1     16,695  00
  07CB     0      0.1         1,113      STK 10030              0      1     16,695  00
```

VERIFY SAMPLE OUTPUT

This example shows the result of a successful VERIFY of a previously erased 3390-9 DASD. All tracks on the volume were verified.

```
FDR001  FDR ERASE VOLUMES         - FDRERASE                - INNOVATION DATA PROCESSING
FDR303  CARD IMAGE --  VERIFY  TYPE=FULL,ERASESTARTCYL=0,ERASESTARTTRK=0,
FDR303  CARD IMAGE --    ERASESKIP=1
FDR303  CARD IMAGE --  MOUNT ERASEUNIT=91C0
FDR235  FDRERASE WILL CHECK THE FOLLOWING     1 UNITS:
FDR235   91C0
FDR170   DEVICE IS ELIGIBLE FOR ERASE UNIT=91C0 VOL=SH91C0 VOLID=FDR5 CU=IBM12345/0801-01000000
00000000
FDR177   VERIFY CHECKED UNIT=91C0 AND ALL TRACKS CONTAINED ERASE DATA PATTERN=FE(RANDOM)
FDR122  OPERATION STATISTICS FOR 3390 VOLUME.................  91C0
FDR122                        CYLINDERS ON VOLUME.............10,017
FDR122                        DASD TRACKS VERIFIED............150,255
FDR122                        BYTES READ FROM DASD.................0
FDR122                        DASD TRACKS ERASED...................0
FDR122                        NUMBER OF ERASE PASSES...............0
FDR122                        DASD EXCPS.....................10,021
FDR122                        TARGET DASD EXCPS....................0
FDR122                        CPU TIME (SECONDS)..............46.980
FDR122                        ELAPSED TIME (MINUTES).............4.4
FDR122                        ERASE TIME.........................4.4
FDR999   FDR SUCCESSFULLY COMPLETED
```

PRINT SAMPLE OUTPUT

This example shows is the result of an FDRERASE PRINT on two tracks of a volume before any data was erased. You can see many data records on each track. There is one line for every record on the track (including Record 0 (R0)). The first and last eight bytes of each record are printed in hex and the first eight bytes are printed in EBCDIC.

```
FDR170   DEVICE IS ELIGIBLE FOR ERASE UNIT=90C2 VOL=SH90C2 VOLID=VOL1 CU=IBM24678/0800-00000000 00000000        17.03.52
FDR175  PRINT UNIT=90C2  CYL.....3 TRK..0 REC...0 KL...0 DL.....8  DATA=0000000000000000 - 15 RECORDS ON TRACK
FDR175  PRINT UNIT=90C2  CYL.....3 TRK..0 REC...1 KL...0 DL..3120  DATA=F0F1F2F3F4F5F6F7.....0000000098000100   01234567
FDR175  PRINT UNIT=90C2  CYL.....3 TRK..0 REC...2 KL...0 DL..3120  DATA=0000543001000000.....C9D5C740C4E2D540   ........
FDR175  PRINT UNIT=90C2  CYL.....3 TRK..0 REC...3 KL...0 DL..3120  DATA=4040404040404040.....0000000000F00000
FDR175  PRINT UNIT=90C2  CYL.....3 TRK..0 REC...4 KL...0 DL..3120  DATA=C1C2C3C4C5C6C7C8.....956B30004770C5BC   ABCDEFGH
FDR175  PRINT UNIT=90C2  CYL.....3 TRK..0 REC...5 KL...0 DL..3120  DATA=4140D1F095404000.....400050004770A1D6   . J0.  .
FDR175  PRINT UNIT=90C2  CYL.....3 TRK..0 REC...6 KL...0 DL..3120  DATA=4040404040404040.....45E0B26A4400C536
FDR175  PRINT UNIT=90C2  CYL.....3 TRK..0 REC...7 KL...0 DL..3120  DATA=0102030405060708.....4200F000D601F000   ........
FDR175  PRINT UNIT=90C2  CYL.....3 TRK..0 REC...8 KL...0 DL..3120  DATA=BCCCDC01F000BCC7.....FF00000000000000   ....0..H
FDR175  PRINT UNIT=90C2  CYL.....3 TRK..0 REC...9 KL...0 DL..3120  DATA=0000FFFFFFFFFFFF.....8A4058308DF04590   ........
FDR175  PRINT UNIT=90C2  CYL.....3 TRK..0 REC..10 KL...0 DL..3120  DATA=8AB458308DF44590.....C2C90102D3D7D401   ........
FDR175  PRINT UNIT=90C2  CYL.....3 TRK..0 REC..11 KL...0 DL..3120  DATA=02D9E2E5010A60F0.....C1C2D9D3D3C9C240   .RSV...0
FDR175  PRINT UNIT=90C2  CYL.....3 TRK..0 REC..12 KL...0 DL..3120  DATA=C6C4D9D6D7E34040.....D41CBD02BF3F307C   FDROPT
FDR175  PRINT UNIT=90C2  CYL.....3 TRK..0 REC..13 KL...0 DL..3120  DATA=4780CDB2D209D41C.....00FB80FA01000000   ....K.M.
FDR175  PRINT UNIT=90C2  CYL.....3 TRK..0 REC..14 KL...0 DL..3120  DATA=0000000000000000.....1001100092F01000   ........
FDR175  PRINT UNIT=90C2  CYL.....3 TRK..0 REC..15 KL...0 DL..3120  DATA=D21E10019216D207.....181D410002344110   K.....K.
FDR175  PRINT UNIT=90C2  CYL.....4 TRK..0 REC...0 KL...0 DL.....8  DATA=0004000000000000 - 15 RECORDS ON TRACK
FDR175  PRINT UNIT=90C2  CYL.....4 TRK..0 REC...1 KL...0 DL..3120  DATA=0000000000000000 - ALL BYTES THE SAME   ........
FDR175  PRINT UNIT=90C2  CYL.....4 TRK..0 REC...2 KL...0 DL..3120  DATA=0000000000000000 - ALL BYTES THE SAME   ........
FDR175  PRINT UNIT=90C2  CYL.....4 TRK..0 REC...3 KL...0 DL..3120  DATA=0000000000000000 - ALL BYTES THE SAME   ........
FDR175  PRINT UNIT=90C2  CYL.....4 TRK..0 REC...4 KL...0 DL..3120  DATA=0000000000000000 - ALL BYTES THE SAME   ........
FDR175  PRINT UNIT=90C2  CYL.....4 TRK..0 REC...5 KL...0 DL..3120  DATA=0000000000000000 - ALL BYTES THE SAME   ........
FDR175  PRINT UNIT=90C2  CYL.....4 TRK..0 REC...6 KL...0 DL..3120  DATA=0000000000000000 - ALL BYTES THE SAME   ........
FDR175  PRINT UNIT=90C2  CYL.....4 TRK..0 REC...7 KL...0 DL..3120  DATA=0000000000000000 - ALL BYTES THE SAME   ........
FDR175  PRINT UNIT=90C2  CYL.....4 TRK..0 REC...8 KL...0 DL..3120  DATA=0000000000000000 - ALL BYTES THE SAME   ........
FDR175  PRINT UNIT=90C2  CYL.....4 TRK..0 REC...9 KL...0 DL..3120  DATA=0000000000000000 - ALL BYTES THE SAME   ........
FDR175  PRINT UNIT=90C2  CYL.....4 TRK..0 REC..10 KL...0 DL..3120  DATA=0000000000000000 - ALL BYTES THE SAME   ........
FDR175  PRINT UNIT=90C2  CYL.....4 TRK..0 REC..11 KL...0 DL..3120  DATA=0000000000000000 - ALL BYTES THE SAME   ........
FDR175  PRINT UNIT=90C2  CYL.....4 TRK..0 REC..12 KL...0 DL..3120  DATA=0000000000000000 - ALL BYTES THE SAME   ........
FDR175  PRINT UNIT=90C2  CYL.....4 TRK..0 REC..13 KL...0 DL..3120  DATA=0000000000000000 - ALL BYTES THE SAME   ........
FDR175  PRINT UNIT=90C2  CYL.....4 TRK..0 REC..14 KL...0 DL..3120  DATA=0000000000000000.....DE805800DE7C0600   ........
FDR175  PRINT UNIT=90C2  CYL.....4 TRK..0 REC..15 KL...0 DL..3120  DATA=5000DE7C5810DE78.....0000000000000000   ........
```

This example shows the same FDRERASE PRINT after the volume was erased with ERASE. You can see that each track contains a single track-length (56664 bytes) record containing all zeros (the default pattern for ERASE). “ALL BYTES THE SAME” indicates that every byte in the data record is identical.

```
FDR170   DEVICE IS ELIGIBLE FOR ERASE UNIT=90C2 VOL=SH90C2 VOLID=FDR5 CU=IBM24678/0800-00000000 00000000        17.11.48
FDR175  PRINT UNIT=90C2  CYL.....3 TRK..0 REC...0 KL...0 DL.....8  DATA=0000000000000000 -  1 RECORDS ON TRACK
FDR175  PRINT UNIT=90C2  CYL.....3 TRK..0 REC...1 KL...0 DL.56664  DATA=0000000000000000 - ALL BYTES THE SAME   ........
FDR175  PRINT UNIT=90C2  CYL.....4 TRK..0 REC...0 KL...0 DL.....8  DATA=0000000000000000 -  1 RECORDS ON TRACK
FDR175  PRINT UNIT=90C2  CYL.....4 TRK..0 REC...1 KL...0 DL.56664  DATA=0000000000000000 - ALL BYTES THE SAME   ........
```

330.8 FDRERASE ISPF INTERFACE

The FDRERASE ISPF interface allows you to initiate, monitor, and control FDRERASE operations on the system where your TSO session is logged on. You can:

- Monitor active erase tasks
- Submit FDRERASE jobs
- Suspend and resume active erase tasks
- Terminate active erase tasks
- Display FDRERASE history
- Select devices by unit address, volume serial number, subsystem ID, subsystem serial number, or SMS storage group

INVOKING THE FDRERASE ISPF DIALOGS

The FDRERASE ISPF dialogs are integrated with the ISPF dialogs of FDR, the INNOVATION DATA PROCESSING FDR DASD management system. If you have installed the FDR ISPF dialogs at a release level matching the level of FDRERASE that you are using, your FDR or ABR main ISPF menu will have an option "E" for FDRERASE. If so, you can skip the invocation of ABRALLOC shown below and use that option to invoke the FDRPAS dialogs.

If you do not have an appropriate level of the FDR dialogs installed, or they are at a lower release level than the level of FDRERASE in use, then use the following procedure:

Go to ISPF option 6, or exit ISPF to TSO READY mode, and issue this command:

```
EXEC 'fdrerase.clist.library(ABRALLOC)'
```

specifying the name of the FDRERASE CLIST library that was installed during the installation (section 380). This CLIST allocates all of the required FDRERASE ISPF libraries and invokes the ABR dialogs (including FDRERASE).

If you issued the ABRALLOC from TSO READY, you see a standard ISPF main menu with an additional option "A" (for ABR). If your installation has modified your ISPF main menu, this may look considerably different than you are normally used to seeing. This ISPF menu looks something like:

ISPF PRIMARY OPTION MENU

```
  Menu  Utilities  Compilers  Options  Status  Help
--------------------------------------------------------------------------------
                          ISPF Primary Option Menu
Option ===> A

0  Settings      Terminal and user parameters           User ID . : DF
1  View          Display source data or listings        Time. . . : 09:05
2  Edit          Create or change source data           Terminal. : 3278
3  Utilities     Perform utility functions              Screen. . : 1
4  Foreground    Interactive language processing        Language. : ENGLISH
5  Batch         Submit job for language processing     Appl ID . : ISR
6  Command       Enter TSO or Workstation commands      TSO logon : V48ISPF
7  Dialog Test   Perform dialog testing                 TSO prefix: DF

8  LM Facility   Library administrator functions        System ID : OS24
9  IBM Products  IBM program development products       MVS acct. : **NONE**

10 SCLM          SW Configuration Library Manager       Release . : ISPF 4.8
11 Workplace     ISPF Object/Action Workplace
A  FDR/ABR       FDR/ABR DASD Management Functions
```

Type "A" on the Option line, as shown, and press ENTER to display the FDR/ABR primary options menu.

FDR PRIMARY OPTIONS MENU

The FDR primary options menu looks like:

FDR PRIMARY OPTIONS MENU – PANEL A

```
---------- FDR TOTAL DASD MANAGEMENT SYSTEM -- FDR PRIMARY OPTIONS MENU ------
OPTION  ===> E

   1  REPORTS     - ABR REPORTING FUNCTIONS
   2  RESTORE     - ABR DATA SET RESTORE
   3  ARCHIVE     - ABR DATA SET ARCHIVE OR SUPERSCRATCH
   4  BACKUP      - ABR DATA SET BACKUP
   5  REMOTE Q    - ABR REMOTE QUEUE UTILITY FUNCTIONS

   C  COMPAKTOR   - COMPAKTOR MAP AND SIMULATION REPORTS
   R  RELEASE     - COMPAKTOR RELEASE

   I  INSTALL     - INSTALLATION AND MAINTENANCE OF FDR AND OPTIONAL PRODUCTS
   J  JCL PARMS   - SPECIFY FDR JCL AND SYSOUT DEFAULTS FOR SUBMITTED JOBS
   K  FORMAT      - MODIFY FORMAT OF GENERATED REPORTS

   M  MESSAGES    - FDR MESSAGES AND CODES QUERY FACILITY
   P  PLUG & SWAP - FDRPAS PLUG & SWAP
   E  FDRERASE    - FDR DISK ERASE
   Q  QUERY       - FDR/ABR STATISTICS QUERY
   S  SRS         - SEARCH, REPORT, SERVICES DIALOG
   T  FDRTSEL     - BACKUP FILE MANAGEMENT UTILITY
```

Select option “E” to monitor and control FDRERASE. Other options are used for FDRABR and other components of the FDR DASD management software. Option “I” is used during installation of FDRPAS. See Chapter 380 “Installation & Maintenance”.

FDRERASE PANEL

This panel is displayed by the "E" option. It is used to initiate, monitor, and control FDRERASE sessions.

FDRERASE – PANEL A.E

```
-------------------------------- FDRERASE ------------------- Row 1 to 1 of 1
COMMAND ===>                                                  SCROLL ===> PAGE

 Active QUick ERase SEcure SIm EMpty ABort SUspend REsume OPtions HIstory SOrt
                                                               Refresh 0
Command  Unit Volume                          Elapsed      Control   SMS
         Addr Serial Cmp Pass                  Time         Unit   Storage
         Mask  Mask   %   #  Status          MM:SS  SSID Serial#  Group   Cyls
-------  ---- ------ --- --- -------------- ------- ---- ------- -------- ----
'''
|-------------------------------------------------------------------------|
| Press enter for Active units OR specify Unit Address, Volser, SSID, Storage|
| Group, or Controller Serial - masking allowed.                          |
|-------------------------------------------------------------------------|
```

If you simply press ENTER, the panel checks to see if there are any FDRERASE tasks in progress on this system. If active tasks were found, they are automatically displayed, such as:

FDRERASE – PANEL A.E WITH ENTER KEY

```
-------------------------------- FDRERASE ------------------- Row 1 to 2 of 2
COMMAND ===>                                                  SCROLL ===> PAGE

 Active QUick ERase SEcure SIm EMpty ABort SUspend REsume OPtions HIstory SOrt
                                                               Refresh 0
Command  Unit Volume                          Elapsed      Control   SMS
         Addr Serial Cmp Pass                  Time         Unit   Storage
         Mask  Mask   %   #  Status          MM:SS  SSID Serial#  Group   Cyls
-------  ---- ------ --- --- -------------- ------- ---- ------- -------- -----
         21C0         12 1/1 ACTIVE  QUICK    0:25  0311                  10017
         21C3         33 1/1 ACTIVE  QUICK    0:19  0311                   3339
```

You have several options:

- As long as there are no volumes displayed on the screen, you can simply press ENTER to display any FDRERASE swaps that are active on this system.
- You can also request that volumes that are not currently involved in an erase be added to the display; the display can be useful to initiate swaps or to verify the current location of selected volumes. Details are later in this section.
- If there are already volumes displayed, type "Active" (or just "A") on the Command line to add any additional active erases to the display.
- If the display currently contains at least one volume that was added to the display because it was active (by pressing ENTER on an empty screen or by typing "Active"), then pressing ENTER again scans for active volumes again and adds any newly active erases to the display.
- If the display currently contains only volumes that were added by specific request, then pressing ENTER simply updates the status of the displayed volumes, and does not check for active erases.

RECOMMENDATION:

If you just keep pressing ENTER, the panel displays all active erases, and adds erases that later became active. Completed erases remain on the display. However, if an erase begins and ends between two presses of ENTER, then the panel never sees the active task and does not add it to the display.

Alternately, you can enter a volume serial number prefix, a unit address prefix, an SSID, subsystem serial number, or an SMS storage group to display all of the volumes/devices selected, whether active or not. Details are later in this section. Pressing ENTER updates the status of the displayed volumes but does not automatically add any active erases of other

devices (unless you type ACTIVE). This is preferable when you are erasing all DASD on a control unit (SSID) or in a range of addresses, since all those DASD volumes are constantly displayed. You can easily see which ones have active erases, have completed, or have not been erased.

The first time you use the FDRERASE dialog, you should enter the OPTIONS command on the command line to change option:

FDRERASE OPTIONS – PANEL A.E WITH OPTIONS COMMAND

```
--------------------------------- FDRERASE - Options ------------------------
COMMAND ===>                                                SCROLL ===> PAGE

Dialog Options:
DISPLAY       ===> OFFLINE  Display disks: ONLINE, OFFLINE or BOTH

Options:
ACTIVETARGET  ===> BYPASS   PPRC or EMC SRDF mirrored disks: BYPASS or PROCESS
ALREADYERASED ===> BYPASS   Disks already erased: BYPASS or PROCESS
CHECKTARGET   ===> YES      Erase only FDRPAS or empty disks: YES or NO
CONFERASE     ===> NO       Issue console confirmation WTOR message: YES or NO
CPYVOLID      ===> NO       Rebuild the VTOC after erase: YES or NO
ERASEPASS     ===> DEFAULT  Number of times to overwrite each track: DEFAULT, N
ERASEPATTERN  ===> 00               Hexadecimal value to overwrite each track
LOGMESS       ===> YES      Issue SYSLOG message at erase completion: YES or NO
MAXTASKS      ===> 64       Number of disks to erase concurrently: 1-64
ONLINE        ===> BYPASS   Disks that are online: BYPASS or VARYOFF
SELTERR       ===> YES      Abend if selected disk is not processed: YES or NO
VARYON        ===> NOAFTER  Vary volume online after erase: AFTER or NOAFTER

CHANGEVOL     ===>          Volume Serial mask to rename volume after erase
CHANGEVOLNUM  ===>          Starting number for sequential mask substitution

VTOCCYL       ===>          New VTOC starting location cylinder: cccc
VTOCTRK       ===>          New VTOC starting location track: hh
VTOCSIZE      ===>          New VTOC size in tracks: 1-9999

Other options:
Maxvols       ===> 255      Maximum number of volumes to generate per jobstep
Interval      ===>          Refresh interval in seconds

Job Statement Information:
  ===> //useridA  JOB (ACCOUNT),'NAME',NOTIFY=userid
  ===> //*
  ===> //*
  ===> //*
   ===> //FDRERASE EXEC PGM=FDRERASE,REGION=0M
  ===> //STEPLIB   DD  DISP=SHR,DSN=FDRERASE.LOADLIB
  ===> //SYSPRINT  DD  SYSOUT=*
  ===> //FDRSUMM   DD  SYSOUT=*
```

You can over type any of the options shown. The values are saved in your ISPF profile for use with all subsequent FDRERASE operations that are initiated by this ISPF user id.

The DISPLAY option controls whether only offline DASD, online DASD, or both, are displayed when you enter a unit mask, SSID, or serial number. The other options on the upper part of the panel correspond to options that can be specified on an FDRERASE control statement. See Section 330.4 "FDRERASE Control Statements" for a description of these options.

Under "Other Options":

- **Maxvols** specifies the maximum number of DASD devices that the panel consolidate into a single FDRERASE step when you start FDRERASE tasks from the dialog.

- **Interval** is the refresh interval, in seconds (default 2), used when you enter a value for Refresh on the main FDRPAS panel. Refresh counts down to zero, every Interval seconds, and automatically refresh the display. The interval allows monitoring FDRERASE operations without constantly pressing ENTER.

The JCL shown at the bottom is used to submit FDRERASE batch jobs from the dialog. Update this with a valid JOB statement and other required JCL. Update the STEPLIB DD statement with the name of the program library containing FDRERASE.

ADDING VOLUMES TO THE DISPLAY

Back on the main panel, you can add additional devices to the display, selecting them by the volume serial, unit address, SMS storage group, SSID (subsystem ID), or control unit serial number. Volume serial and storage group selects only online volumes; the other options display online DASD, offline DASD, or both, depending on the setting on the option panel.

On a blank line (or even a line that currently displays a DASD volume if you no longer want to see it), tab to the field that you wish to use for selection and enter the appropriate value. If a complete volume serial or unit address is entered, only that one DASD volume is displayed; volume serial can only be used to select DASD that are currently online. If you enter an SMS storage group, SSID, or serial number, all DASD that match the value will be displayed. The SSID or serial number is a convenient way to display all volumes in an unused DASD subsystem so that you can erase all or some of them.

For example, if you enter an SSID:

FDRERASE – PANEL A.E – ADDING VOLUMES TO DISPLAY

```
Command  Unit Volume                          Elapsed      Control   SMS
         Addr Serial Cmp Pass                  Time         Unit   Storage
         Mask  Mask   %   #  Status           MM:SS  SSID Serial#  Group   Cyls
-------  ---- ------ --- --- -------------- ------- ---- ------- -------- -----
'''                                                  0300
```

and press ENTER, the panel displays all volumes in the DASD subsystem with that SSID (online, offline, or both). As many as fits on the screen are displayed and you may need to scroll up and down (PF7 and PF8 or the UP and DOWN commands) to see the entire list.

Wild card characters can be used to select multiple volumes, units, storage groups, or control units. An asterisk (*) is the only supported wild card character. If an asterisk appears in the middle of a selection string it represents exactly one character, while if it appears at the end it represents one or more characters.

For example,

- A volume serial number mask of **DB* selects all online volumes with serials of *xx*DB*xx*
- A unit address mask of 12* selects all DASD with addresses of 12*xx*
- A storage group name of DB**3 selects all online volumes in SMS storage groups with names of DB*xx*3
- An SSID of *3* selects all DASD in subsystems with an SSID of *x*3*xx*

ISPF line commands are supported for inserting and deleting entries in the display. In the “Command” column, next to any entry, enter:

- **D** to delete a volume from the display
- **I** to insert a blank entry in the display; it can be modified to add more volumes to the displayed volume list.

MONITORING FDRERASE

When active erases are displayed, you can monitor the erase progress. The display looks like:

FDRERASE – PANEL A.E – MONITOR PROGRESS

```
Command  Unit Volume                          Elapsed      Control   SMS
         Addr Serial Cmp Pass                  Time         Unit   Storage
         Mask  Mask   %   #  Status           MM:SS  SSID Serial#  Group   Cyls
-------  ---- ------ --- --- -------------- ------- ---- ------- -------- -----
         21C3         33 1/1 ACTIVE  ERASE     0:42  0311                  3339
```

The status of **ACTIVE** indicates that the erase is in progress; the following text indicates the type of erase (**QUICK**, **ERASE**, or **SECURE**). The status can also show **ERASED**, **ERROR** (if the erase failed), **SUSPEND**, or **INACTIVE** (no erase was ever started for the volume).

Other fields show the percentage complete, pass number, elapsed time, and description of the device. You can scroll left and right (PF10 and PF11) to display some additional fields (such as cylinders/second).

Whenever you press ENTER, the information displayed is updated. Alternately, you can over type the number after **REFRESH** with a value. The screen is updated automatically, every two seconds, until this value counts down to zero; the two-second refresh interval can be changed on the FDRERASE options panel shown previously in this section.

SORTING AND FILTERING THE DISPLAY

Enter the SORT command on the COMMAND line to sort the displayed volumes and/or filter the display based on STATUS values. The SORT command displays this selection screen:

FDRERASE SORT – PANEL A.E WITH SORT COMMAND

```
-------------------------------- FDRERASE - Sort ---------------------------
COMMAND ===>                                                 SCROLL ===> PAGE


1) Specify the numeric sequence and order (A/D) of field(s) to sort:

   Field    Seq Order  Description
   -------- --- -----  ------------------
   VOL                 Volume  Serial
   UNIT                Unit Address
   SSID                SSID
   STORGRP             Storage Group
   CU                  Controller Serial
   DEVTYPE             Device Type
   STATUS              Status

2) Select the STATUS values of rows to be included in the display:

   Select   Status Value
   -------- ----------------------------
      S     ACTIVE
      S     ERASE
      S     ERASE  *ERROR*
      S     ERASE *CANCEL*
      S     SUSPEND
      S     INACTIVE
```

In the first section, place a digit in the SEQ field for each field on which the display is to be sorted; the field with SEQ of 1 is sorted first, then the field with SEQ of 2, and so on. Place an “A” (ascending) or “D” (descending) in the ORDER field to specify the sort order (“A” is assumed if not specified). Note that sorting on the STATUS field, ascending, places ACTIVE erases first.

INNOVATION DATA PROCESSING suggests that sorting on STATUS (1) and VOL (2) will provide a useful view of your volumes, especially if you have selected a range of volsers, device addresses, or an SSID to display.

In the second section, blank out the “S” in front of any values for STATUS that you do not want to see. Only those volumes with a status for which “S” is still present is displayed when you press END (PF3) to return to the status screen. If one or more values have the “S” blanked out, the status screen shows the text “*ROWS EXCLUDED*” to remind you that you have filtered the display.

The values you fill in on this screen will be remembered in your ISPF profile and used the next time you re-enter the FDRPAS dialogs. To change the values, you must use the SORT command again and modify the values.

INITIATING ERASES

You can also initiate an erase from the FDRPAS ISPF panels, by submitting a batch job. Before you do so, you must update the JCL parameters on the dialog OPTIONS panel, as shown earlier. You may want to update other options on that panel. If you are going to erase volumes that are currently online, you must set the ONLINE option to VARYOFF.

FDRERASE – PANEL A.E – INITIATE ERASE JOBS

```
-------------------------------- FDRERASE ------------------- Row 1 to 9 of 9
COMMAND ===>                                                  SCROLL ===> PAGE

 Active QUick ERase SEcure SIm EMpty ABort SUspend REsume OPtions HIstory SOrt
                                                              Refresh 0
Command  Unit Volume                          Elapsed      Control   SMS
         Addr Serial Cmp Pass                  Time         Unit   Storage
         Mask  Mask   %   #  Status           MM:SS  SSID Serial#  Group   Cyls
-------  ---- ------ --- --- -------------- ------- ---- ------- -------- -----
erase    21CD               INACTIVE               0311          3390     1113
quick    21CC               INACTIVE               0311          3390     1113
```

To initiate an erase, display one or more inactive DASD devices (not currently involved in an erase) as described earlier. You can enter **SIM** (SIMERASE), **QUICK** (QUICKERASE), **ERASE**, **SECURE** (SECURERASE), or **EMPTY** (EMPTYVTOC):

- ❖ Enter the command in the "Command" column next to each volume to be swapped as shown above.
- ❖ Enter the command on the "COMMAND" line at the top of the screen. The "COMMAND" line will apply to every volume on the display (even those that may not be currently visible) that is in INACTIVE status.

Entering a command creates one or more batch jobs to erase the selected DASD. You have the option to browse or edit the generated job stream, or submit it immediately.

NOTE: While QUICKERASE is supported, it is not documented nor recommended for use since it does not meet the certified methods of erasing DASD.

SUSPENDING AND RESUMING ACTIVE ERASES

You can use the FDRERASE ISPF panel to temporarily suspend an active erase without terminating it and resume it later.

Enter the "SUSPEND" command (or just SU) in the "Command" column next to the active erase to be suspended. The status will change to SUSPEND. To resume, enter the "RESUME" command (or just RE) next to any suspended swap.

FDRERASE – PANEL A.E – SUSPEND ACTIVE ERASE

```
Command  Unit Volume                          Elapsed      Control   SMS
         Addr Serial Cmp Pass                  Time         Unit   Storage
         Mask  Mask   %   #  Status           MM:SS  SSID Serial#  Group   Cyls
-------  ---- ------ --- --- -------------- ------- ---- ------- -------- -----
suspend  21C0         12 1/1 ACTIVE  ERASE    0:19  0311                  10017
```

TERMINATING ACTIVE ERASES

You can use the FDRERASE ISPF panel to terminate an active erase, if the need arises.

Enter the "ABORT" command (or just AB) in the "Command" column next to the active erase to be aborted.

FDRERASE – PANEL A.E – TERMINATE ACTIVE ERASE

```
Command  Unit Volume                          Elapsed      Control   SMS
         Addr Serial Cmp Pass                  Time         Unit   Storage
         Mask  Mask   %   #  Status           MM:SS  SSID Serial#  Group   Cyls
-------  ---- ------ --- --- -------------- ------- ---- ------- -------- -----
abort    21C3         33 1/1 ACTIVE  ERASE    0:19  0311                   3339
```

DISPLAYING ERASE HISTORY

If you enter the “HISTORY” command (or just HI) on the command line of the FDRERASE panel, you can display FDRERASE history records, as shown above, for all volumes that have been successfully erased on this system since the last IPL.

FDRERASE – PANEL A.E – HISTORY COMMAND

```
-------------------------------- FDRERASE - History --------- Row 1 to 4 of 4
COMMAND ===>                                                  SCROLL ===> PAGE
  HIstory SOrt

Command  Unit Volume                        Elapsed      Control   SMS
         Addr Serial Cmp Pass                Time         Unit   Storage
         Mask  Mask   %   #  Status          MM:SS  SSID Serial#  Group   Cyls
-------  ---- ------ --- --- -------------- ------- ---- ------- -------- -----
        330C SY330C      1/1  ERASED  ERASE    2:53  3004 00547            3339
        330E SY330E      1/1  ERASED  ERASE    2:47  3004 00547            3339
        23C5             1/1  ERASED  ERASE    0:42  8103 76421            3339
        23C6             1/1  ERASED  ERASE    0:41  8103 76421            3339
```

If you enter a device address or an address mask (for example, 21*) after the HISTORY command, only those devices are displayed.

THIS PAGE INTENTIONALLY LEFT BLANK

380 INSTALLATION & MAINTENANCE

380.1 LOADING THE INSTALLATION LIBRARIES

FDRPAS, FDRMOVE, and FDRERASE can be installed and executed on any release of z/OS.

The installation of FDRPAS, FDRMOVE, and/or FDRERASE is done in two different ways:

1. **Electronic Installation** – The installation package is downloaded from the INNOVATION DATA PROCESSING FTP site. If you are installing the products using this method, see Section 380.2 "Electronic Installation".
2. **CD Installation** – The installation package is downloaded from a CD. If you are installing the products using this method, see Section 380.3 "CD Installation".

380.2 ELECTRONIC INSTALLATION

ELECTRONIC INSTALLATION STEP 1.1

With the electronic installation, instructions are sent via a set of two e-mails for installing the libraries. The e-mail installation instructions supersede these printed instructions and should be used if any differences are noted.

The order acknowledgment e-mail (first of two e-mails) looks similar to:

```
Subject: Innovation Distribution Server Order Acknowledgment 012345
From: ENSMTP@CPUA.IDPNJ.COM
To: SYSPROG@ABC.COMPANY

You are receiving the first of a two message set acknowledging your Innovation
software product order. Retain this e-mail until you receive a status (second)
message containing instructions for performing the product installation. The
status message will be sent when your order is ready to be downloaded from the
Innovation FTP site.

NOTE:
   1. You must perform the install of this distribution file
   before THURSDAY, NOVEMBER 28 2013.
   The install program IDPREC in your distribution file will
   expire on this day and the file will be removed from our
   FTP site.

This order is for PAS Product Group including the following options:
  FDRPAS FDRERASE FDRMOVE
  RENTAL CUSTOMER LICENSE

Customer Order Number:
  012345012345012345

Company name:
  ABC COMPANY

Customer contact:
  SYSPROG

Product version:        Rental Date:
  5.4/80                 2014.060

IDPREC Password:
  012345012345012345

************************************************************
This message came to you from the Innovation Distribution
Server. You can respond to the sender of this message, or
send questions by Email to: support@fdrinnovation.com
************************************************************
This email and any files transmitted with it are confidential
and intended solely for the individual or entity to whom they
are addressed. If you have received this email in error
contact Innovation at support@fdrinnovation.com
************************************************************
Copyright 2006-2013 INNOVATION DATA PROCESSING
************************************************************
```

This first e-mail contains the products and versions that are included in the installation package, the expiration date (if the product has an expiration date), and a unique IDPREC password that is required for the installation process.

ELECTRONIC INSTALLATION STEP 1.2

The order completion e-mail (second of two e-mails) looks similar to:

```
Subject: Innovation Distribution Server Order Completion 012345
From: ENSMTP@CPUA.IDPNJ.COM
To: SYSPROG@ABC.COMPANY
```

```
You are receiving the second of a two message set acknowledging your Innovation
software product order is ready to be downloaded from the Innovation FTP site.
Verify that the Customer Order Number is the same on both e-mail messages before
performing the retrieval procedure below.

This order is for PAS Product Group including the following options:
  FDRPAS FDRERASE FDRMOVE
  RENTAL CUSTOMER LICENSE

Customer Order Number:
  012345012345012345

Customer number:
  012345

Company name:
  ABC COMPANY

Customer contact:
  SYSPROG

Product version:       Expiration Date:
  5.4/80                 2014.060

The instructions below outline the procedure you will use to retrieve the product
file from the Innovation FTP site and then install our software.

The most convenient method is to …
```

This second e-mail contains the links to a unique FTP site containing the product installation package. The transfer of the installation package can be done with a Java applet or via manual FTP download if you encounter problems running the Java applet at your site. Instructions are included in this e-mail for both methods.

ELECTRONIC INSTALLATION STEP 1.3

The Java applet installation procedure is the easiest and downloads a product distribution file from the INNOVATION DATA PROCESSING FTP site to your PC by using the Windows FTP.EXE client. It then connects to your z/OS FTP server to transfer this file to z/OS. This process is done within the user's local network; data is never transmitted across the internet. You needs to supply your z/OS connection information and click the button labeled “Start Transfer”.

Upon successful file transfer to your z/OS host, the applet displays a confirmation message box labeled “Transfer to MVS Host Successful”. After clicking “OK”, another message box containing the remaining z/OS installation instructions is displayed.

ELECTRONIC INSTALLATION STEP 1.4

Follow the "Remaining Product Installation Instructions" that are presented at the end of the file transfer. Expanding the product distribution file on your z/OS host is a two-step process. If you are using ISPF, issue the following TSO commands from ISPF Option 6 (TSO Commands). You can also exit ISPF and issue them from the TSO "READY" prompt. This TSO session must have access to an ISPF environment because the z/OS installation process expects to employ ISPF panels to complete the installation. The steps described below assume TSO PROFILE NOPREFIX is NOT used.

1. Enter:

   ```
   RECEIVE INDATASET(PAS.BIN)
   ```

   Where the INDATASET(…) value is the name that you created during the file transfer to z/OS.

   This creates a partitioned data set containing two load modules. After the RECEIVE is completed, the name of the PDS is "*user id*.IDPREC.LOAD". If you want a different name, then at the prompt:

   ```
   Data set FDRSYS.IDPREC.LOAD from FDRCFG on JESCPUA
   Enter restore parameters or 'DELETE' or 'END' +
   ```

   Enter:

   ```
   DSN(desired.name)
   ```

   Or take the default by pressing **ENTER**.

2. Enter:

   ```
   CALL 'userid.IDPREC.LOAD(IDPREC)'
   ```

   or, if you gave the PDS a different name:

   ```
   CALL 'dsname(IDPREC)'
   ```

   A welcome message is displayed. Enter appropriate responses to name and SYSOUT class prompts.

   You are also prompted to enter the 16-digit IDPREC PASSWORD supplied in the Order Acknowledgment e-mail

   Follow the instructions on the ISPF panel to complete the installation of the distribution files.

ELECTRONIC INSTALLATION STEP 1.5: DATA SET NAME SELECTION

This screen allows you to specify the data set names that are used for the data sets you are loading from the installation package. These may be existing data sets to be updated, or they may be new data sets that are allocated and cataloged (new data sets are recommended). Do not install FDRPAS, FDRMOVE, and/or FDRERASE into existing libraries containing other FDR products, since they may share common modules.

IDP PRODUCT INSTALL – SCREEN 1

```
------------------------- IDP Product Install (Screen1) ------------------
  Command  ===>

    Press  Enter  to accept changes
    Press  End    to continue install
    Enter  Can        to quit

  Product  ==> PAS      Version  ==> 5.4/80
  Input  DSN (from FTP site)     ==> IDP.PAS.SHIPPDS
  Output DSN:
   High Level Qualifier          ==> IDP
   2nd Level Qualifier           ==> PAS5480
   Low Level Qualifiers:  ('X ' Will Bypass File Processing)
    CLIST    ==> CLIST                SKELETON  ==> SKELETON
    LOADLIB  ==> LOAD                 TABLES    ==> NOT.SHIPPED
    MESSAGES ==> MESSAGES             ICL       ==> ICL
    PANELS   ==> PANELS               JCL       ==> JCL


   STORCLAS  ==>            MGMTCLAS  ==>
   VOLUME    ==>            UNIT      ==>

  Resulting Data Set Names
  CLIST     ==> IDP.PAS5480.CLIST
  LOADLIB   ==> IDP.PAS5480.LOAD
  MESSAGES  ==> IDP.PAS5480.MESSAGES
  PANELS    ==> IDP.PAS5480.PANELS
  SKELETON  ==> IDP.PAS5480.SKELETON
  TABLES    ==> ------------------------
  ICL       ==> IDP.PAS5480.ICL
  JCL       ==> IDP.PAS5480.JCL
```

The load library must be an APF authorized library. If necessary, you can authorize it after the installation by using the SETPROG console command (see the IBM *z/OS MVS System Commands (SA22-7627)* manual for details).

The names shown above are the default names provided by the IDPREC installation program (when the default user of IDP being used). You may change these names in one of two ways:

1. To change the High or 2nd Level qualifiers for all data sets, change the values in the corresponding field for the Output DSN section.
2. To change an individual Low Level qualifier, change desired Low Level Qualifier for the data set(s) that you want to change.

The resulting data set names that are used are displayed at the bottom area of the panel.

You can also specify the SMS attributes (if they are to be SMS-managed) or VOLUME/UNIT information on this panel to allocate these data sets when creating them as NEW.

ELECTRONIC INSTALLATION SCREEN2 – IDP PRODUCT INSTALLATION

This screen documents the process that is used to create the product libraries. When you press ENTER to continue, it shows you the status of each of the data sets being created. A completion message is displayed once all the product libraries have been created.

IDP PRODUCT INSTALL – SCREEN 2

```
------------------------- IDP Product Install (Screen2) ------------------
 Command  ===>
   Press  Enter  to continue
   Enter  Can/Cancel to quit


     The TSO RECEIVE command will be invoked to create each file
     selected from the previous menu. Before each file is created
     by using TSO RECEIVE the target data set will be deleted.
     The install will display messages below indicating install
     progress and results. If a file RECEIVE fails the user is given
     the option to continue or quit. The user can view RECEIVE output
     by looking at the RECEIVE sysout allocation under their TSO user
     session. If any dialog file fails to RECEIVE it is suggested that
     the user NOT continue to the next screen.

     The next step will change member ABRALLOC in the Clist library
     that is used to allocate libraries and invoke the product dialog.
```

ELECTRONIC INSTALLATION STEP 1.6: INSTALLATION DATA SETS HAVE BEEN RECEIVED

This screen documents that all the product libraries have been created and received and goes to the Primary Options Menu to continue the installation process and customize the product. You can exit the menu at this time as all the necessary libraries have been loaded.

IDP PRODUCT INSTALL – SCREEN 3

```
------------------------- IDP Product Install (Screen3) ------------------
  Command  ===>
    Press  Enter  to continue
    Enter  Can/Cancel to quit


      The  PAS       installation data sets have been received.

      If you would like to display the FDR PRIMARY OPTIONS MENU
      and continue with the Installation process, press ENTER;
      otherwise press CANCEL.
```

Proceed to Section 380.5 "Invoking the Installation ISPF Dialog (Panel A.I)" to complete the installation.

380.3 CD INSTALLATION

CD INSTALLATION STEP 1

The CD installation method contains a letter that documents the installation process. (A copy of this letter is in electronic form in the **Readme.rtf** file in the zOS directory found in the root directory of the CD). You should follow these instructions provided to install the product by using the CD installation method. You must transfer the PAS.BIN file from the zOS directory found in the root directory of the CD to your z/OS host. The required file attributes for the target z/OS data set is listed in the instructions. The exact directions for the FTP transfer are also documented in the instructions.

CD INSTALLATION STEP 2

Once the PAS.BIN file is on the z/OS host system, you then expand the product distribution file. Expanding the product distribution file on your z/OS host is a two-step process. If you are using ISPF, issue the following TSO commands from ISPF Option 6 (TSO Commands). You can also exit ISPF and issue them from the TSO "READY" prompt. This TSO session must have access to an ISPF environment because the z/OS installation process expects to employ ISPF panels to complete the installation. The steps described below assume TSO PROFILE NOPREFIX is NOT used.

1. Enter:

   ```
   RECEIVE INDATASET(PAS.BIN)
   ```

   Where the INDATASET(…) value is the name that you created during the file transfer to z/OS.

   This creates a partitioned data set containing two load modules. After the RECEIVE is completed, the name of the PDS is "*user id*.IDPREC.LOAD". If you want a different name, then at the prompt

   ```
   Data set FDRSYS.IDPREC.LOAD from FDRCFG on JESCPUA
   Enter restore parameters or 'DELETE' or 'END' +
   ```

   Enter:

   ```
   DSN(desired.name)
   ```

   Or take the default by pressing ENTER.

2. Enter:

   ```
   CALL 'userid.IDPREC.LOAD(IDPREC)'
   ```

   or, if you gave the PDS a different name:

   ```
   CALL 'dsname(IDPREC)'
   ```

   A welcome message is displayed. Enter appropriate responses to name and SYSOUT class prompts.

   You are also prompted to enter the 16-digit IDPREC PASSWORD supplied in the Order Acknowledgment e-mail.

   Follow the instructions on the ISPF panel to complete the installation of the distribution files.

CD INSTALLATION STEP 3 DATA SET NAME SELECTION

This screen allows you to specify the data set names that are used for the data sets you are loading from the installation package. These may be existing data sets to be updated, or they may be new data sets that are allocated and cataloged (new data sets are recommended). Do not install FDRPAS, FDRMOVE, and/or FDRERASE into existing libraries containing other FDR products, since they may share common modules.

IDP PRODUCT INSTALL – SCREEN 1

```
------------------------- IDP Product Install (Screen1) ------------------
 Command  ===>

   Press  Enter  to accept changes
   Press  End    to continue install
   Enter  Can        to quit

 Product  ==> PAS      Version  ==> 5.4/80
 Input  DSN (from FTP site)     ==> IDP.PAS.SHIPPDS
 Output DSN:
  High Level Qualifier          ==> IDP
  2nd Level Qualifier           ==> PAS5480
  Low Level Qualifiers:  ('X ' Will Bypass File Processing)
   CLIST    ==> CLIST                SKELETON  ==> SKELETON
   LOADLIB  ==> LOAD                 TABLES    ==> NOT.SHIPPED
   MESSAGES ==> MESSAGES             ICL       ==> ICL
   PANELS   ==> PANELS               JCL       ==> JCL


  STORCLAS  ==>            MGMTCLAS  ==>
  VOLUME    ==>            UNIT      ==>

 Resulting Data Set Names
 CLIST     ==> IDP.PAS5480.CLIST
 LOADLIB   ==> IDP.PAS5480.LOAD
 MESSAGES  ==> IDP.PAS5480.MESSAGES
 PANELS    ==> IDP.PAS5480.PANELS
 SKELETON  ==> IDP.PAS5480.SKELETON
 TABLES    ==> ------------------------
 ICL       ==> IDP.PAS5480.ICL
 JCL       ==> IDP.PAS5480.JCL
```

The load library must be an APF authorized library. If necessary, you can authorize it after the installation by using the SETPROG console command (see the IBM *z/OS MVS System Commands (SA22-7627)* manual for details).

The names shown above are the default names provided by the IDPREC installation program (when the default user of IDP being used). You may change these names in one of two ways:

1. To change the High or 2nd Level qualifiers for all data sets, change the values in the corresponding field for the Output DSN section.
2. To change an individual Low Level qualifier, change the desired Low Level Qualifier for the data set(s) that you want to change.

The resulting data set names that are used are displayed at the bottom area of the panel.

You can also specify the SMS attributes (if they are to be SMS-managed) or VOLUME/UNIT information on this panel to allocate these data sets when creating them as NEW.

CD INSTALLATION IDP PRODUCT INSTALLATION

This screen documents the process that is used to create the product libraries. When you press ENTER to continue, it shows the status of each of the data sets being created. A completion message is displayed once all the product libraries have been created.

IDP PRODUCT INSTALL – SCREEN 2

```
-------------------------- IDP Product Install (Screen2) ------------------
 Command  ===>
   Press  Enter  to continue
   Enter  Can/Cancel to quit


     The TSO RECEIVE command will be invoked to create each file
     selected from the previous menu. Before each file is created
     by using TSO RECEIVE the target data set will be deleted.
     The install will display messages below indicating install
     progress and results. If a file RECEIVE fails the user is given
     the option to continue or quit. The user can view RECEIVE output
     by looking at the RECEIVE sysout allocation under their TSO user
     session. If any dialog file fails to RECEIVE it is suggested that
     the user NOT continue to the next screen.

     The next step will change member ABRALLOC in the Clist library
     that is used to allocate libraries and invoke the product dialog.
```

CD INSTALLATION STEP 4 INSTALLATION DATA SETS HAVE BEEN RECEIVED

This screen documents that all the product libraries have been created and received and goes to the Primary Options Menu to continue the installation process and customize the product. You can exit the menu at this time as all the necessary libraries have been loaded.

IDP PRODUCT INSTALL – SCREEN 3

```
-------------------------- IDP Product Install (Screen3) ------------------
 Command  ===>
   Press  Enter  to continue
   Enter  Cancel to quit


     The  PAS       installation data sets have been received.

     If you would like to display the FDR PRIMARY OPTIONS MENU
     and continue with the Installation process, press ENTER;
     otherwise press CANCEL.
```

Proceed to Section 380.5 "Invoking the Installation ISPF Dialog (Panel A.I)" to complete the installation.

380.5 INVOKING THE INSTALLATION ISPF DIALOG (PANEL A.I)

The CLIST Library contains a member named ABRALLOC for installation and testing of the new FDRPAS/FDRMOVE/FDRERASE system. This CLIST invokes the ISPF dialogs by concatenating the ISPF libraries in front of your current ISPF library allocations. During the tape installation process, the CLIST library was updated to reflect all of the library names where you loaded the new version.

If you installed by another method (such as from the INNOVATION DATA PROCESSING FTP site), you may need to edit member ABRALLOC in the CLIST library to specify the library names you used.

ISPF DIALOG INSTALLATION STEP 1

Issue the following command either under ISPF option "6" (TSO command processor), or under TSO "READY" mode:

```
EXEC 'fdrpas.clist.library(ABRALLOC)'
```

- Use the data set name of the CLIST Library that was specified in the installation process.

This command allocates the FDR ISPF dialog libraries. If you issue the command under ISPF, skip step 2 and proceed to ISPF Dialog Installation Step 3 – FDR Primary Options Menu.

ISPF DIALOG INSTALLATION STEP 2 – ISPF PRIMARY OPTION MENU

A standard ISPF primary menu for your version of ISPF, with the FDR/ABR option added, is displayed. Select option "A" (FDR/ABR) on this menu to access the FDR dialog.

ISPF PRIMARY OPTION MENU

```
 Menu  Utilities  Compilers  Options  Status  Help
--------------------------------------------------------------------------------
                          ISPF Primary Option Menu
Option ===>

0  Settings      Terminal and user parameters          User ID . : USER1
1  View          Display source data or listings       Time. . . : 17:02
2  Edit          Create or change source data          Terminal. : 3278
3  Utilities     Perform utility functions             Screen. . : 1
4  Foreground    Interactive language processing       Language. : ENGLISH
5  Batch         Submit job for language processing    Appl ID . : ISP
6  Command       Enter TSO or Workstation commands     TSO logon : V59ISPF
7  Dialog Test   Perform dialog testing                TSO prefix: USER1
8  LM Facility   Library administrator functions       System ID : CPUB
9  IBM Products  IBM program development products      MVS acct. : **NONE**
10 SCLM          SW Configuration Library Manager      Release . : ISPF 5.9
11 Workplace     ISPF Object/Action Workplace
A  FDR/ABR       FDR/ABR DASD Management Functions

Enter X to Terminate using log/list defaults
```

ISPF DIALOG INSTALLATION STEP 3 – FDR PRIMARY OPTIONS MENU

Select option "I" (INSTALL) on the FDR Primary Options Menu to invoke the installation dialog, as illustrated in the following figure:

FDR PRIMARY OPTIONS MENU – PANEL A

```
---------- FDR TOTAL DASD MANAGEMENT SYSTEM -- FDR PRIMARY OPTIONS MENU ------
OPTION  ===>
                                                                        V 5.4/80
  20  COPY         - FDRCOPY COPY DATA SET FUNCTION
  21  MOVE         - FDRCOPY MOVE DATA SET FUNCTION
  30  SIMREORG     - SIMULATE FDRREORG DATA SET REORGANIZATION FUNCTION
  31  FDRREORG     - FDRREORG DATA SET REORGANIZATION FUNCTION
   O  OPTIONS      - SET OPTIONS AND DEFAULTS FOR FUNCTIONS ABOVE

   C  COMPAKTOR    - COMPAKTOR MAP AND SIMULATION REPORTS
   R  RELEASE      - COMPAKTOR RELEASE

   I  INSTALL      - INSTALLATION AND MAINTENANCE OF FDR AND OPTIONAL PRODUCTS
   J  JCL PARMS    - SPECIFY FDR JCL AND SYSOUT DEFAULTS FOR SUBMITTED JOBS

   P  PLUG & SWAP  - FDRPAS PLUG & SWAP
   E  FDRERASE     - FDR DISK ERASE
   M  FDRMOVE      - FDRMOVE PLUG & SWAP DATA SET MOVE

   MC MESSAGES     - FDR MESSAGES AND CODES QUERY FACILITY
   Q  QUERY        - FDR/ABR STATISTICS QUERY
   T  FDRTSEL      - BACKUP FILE MANAGEMENT UTILITY
```

The installation dialog includes options that are used during the installation and customization of other FDR programs, such as FDR, FDRABR, FDRREORG, and COMPAKTOR. Some of these options also apply to FDRPAS, FDRMOVE, and FDRERASE and are documented in this manual. Others do not apply; they are not shown in this manual and should not be used.

380.13 SET THE FDR GLOBAL OPTIONS TABLE

New Installations:

NOTE: You want to display each of the option panels that apply to FDRPAS, FDRMOVE, and FDRERASE to review and set options appropriately. INNOVATION DATA PROCESSING suggests that you review the text in the following sections of the manual as you go through this process, but you may also display the help panels for any option panel, with much of the same information.

Existing Installations:

NOTE: You should copy the options and tables from your production version of FDRPAS, FDRMOVE, and FDRERASE, but you may wish to review the option panels afterwards to be sure that any new options and options for new components are properly set.

Many options may be changed permanently. Most such options are kept in a load module called FDROPT, in the FDRPAS/FDRMOVE/FDRERASE program library.

There are two ways of modifying these options and tables. The ISPF dialogs may be used to set all options and tables. This is the preferred way, since all options are displayed with online help to describe them. If you have not installed the dialogs or prefer not to use them, the options and tables may be set with program FDRZAPOP, the Global Option Change facility, described in Section 91 in the FDR manual.

PANEL A.I – INSTALLATION OPTIONS MENU

To enter the FDR dialog to display and set options in FDROPT, select option "4" (SETOPT) in the FDR Installation Options Menu, as illustrated in the following figure:

INSTALLATION OPTIONS MENU – PANEL A.I

```
---------- FDR TOTAL DASD MANAGEMENT SYSTEM -- INSTALLATION OPTIONS MENU -----
OPTION  ===>

   1  INSTALL  - LOAD SELECTED LIBRARIES FROM THE FDR DISTRIBUTION TAPE
   1A DSNAMES  - DISPLAY/CHANGE THE DATA SET NAMES OF THE FDR LIBRARIES
   2  LOADSAR  - LOAD THE STAND-ALONE PROGRAM (SAR) ONTO DISK

   4  SETOPT   - SET INSTALLATION OPTIONS IN THE FDR GLOBAL OPTIONS TABLE
   4A DYNAM    - DISPLAY THE DYNAMICALLY INSTALLED FDR GLOBAL OPTIONS TABLE

   5  SETCPK   - SET UP THE COMPAKTOR UNMOVABLE TABLE
   5A SETREORG - SET UP THE FDRREORG NOREORG LIST

   ABR OPTION INSTALLATION

   6  SETLIST  - SET UP THE ABR PROTECT LISTS AND RESTORE ALLOCATION LIST
   7  ABRCAT   - CREATE THE ABR CATALOG
   8  ABRVOL   - SET ABR DISK VOLUME PROCESSING OPTIONS
   9  BLDARC   - CREATE THE ARCHIVE CONTROL FILE
  10  DIALOG   - SET FDR DIALOG GLOBAL OPTIONS

  11  ADDISPF  - ADD FDR COMMANDS TO AN ISPF COMMAND TABLE
```

FDR GLOBAL OPTIONS TABLE

The FDR Global Options Table (module FDROPT) contains installation options for security features, user exits, control statement defaults, and the rest. Options are organized by option type on the Global Options Primary Menu; each option on this panel takes you to another panel where you may display and modify the actual options. Since FDRPAS, FDRMOVE, and FDRERASE are in a separate library from any other FDR products that you are licensed, this procedure modifies only the options used with FDRPAS, FDRMOVE, and FDRERASE.

PANEL A.I.4 – SET FDR GLOBAL OPTIONS PRIMARY MENU

The name of the FDRPAS, FDRMOVE, and FDRERASE program library used during installation is displayed. If necessary, correct that library name; you can also specify the volume serial of the library if it is not cataloged. These values are saved in your ISPF profile so that you do not need to reenter them in the future. The subsequent panels display the options currently in effect in the FDROPT module in that library, and update that library when the options are saved.

SET FDR GLOBAL OPTIONS PRIMARY MENU – PANEL A.I.4

```
---------- FDR INSTALLATION -- SET FDR GLOBAL OPTIONS PRIMARY MENU -----------
OPTION  ===>

    1  - SECURITY OPTIONS                 8  - MORE ABR GENERAL OPTIONS
    2  - GENERAL OPTIONS                  9  - ABR REPORT DEFAULTS
    3  - COMPAKTOR OPTIONS               10  - MORE ABR REPORT DEFAULTS
    4  - ABR GENERAL OPTIONS             11  - OPERATING SYSTEM EXITS
    5  - ABR DATA SET NAMES              12  - FDRREORG OPTIONS
    6  - ABR ARCHIVE UTILITY DEFAULTS    13  - RESERVED
    7  - ABR DISK PROCESSING OPTIONS     14  - FDRPAS & FDRERASE OPTIONS

  SAVE - SAVE OPTION CHANGES            COPY - COPY OPTIONS FROM A PRIOR LEVEL
CANCEL - EXIT WITHOUT SAVING CHANGES    AUDIT - DISPLAY USER CHANGED OPTIONS
REFRESH - REFRESH OPTIONS TABLE IN LPA  RESET - RE-INITIALIZE ALL OPTIONS

 FDR PROGRAM LIBRARY DATA SET:
   DATA SET NAME    ===> 'IDP.PAS5480.LOAD'
   VOLUME SERIAL    ===>

NOTE: TO REFRESH THE OPTIONS THAT ARE DYNAMICALLY INSTALLED IN THE ACTIVE LPA,
IT IS NECESSARY TO RUN FDRSTART - USE THE REFRESH CMD TO GENERATE FDRSTART JCL.
```

SET OPTION COMMANDS

As shown, several special commands are available on this panel:

SAVE – Updates the FDROPT module in the specified program library. No changes are made to the library until "SAVE" is entered, so you may freely switch between option panels and change options until you are satisfied.

CANCEL – Exits to the previous menu and discards all options changed since the last "SAVE" command.

REFRESH – Is not used with FDRPAS, FDRMOVE, and FDRERASE.

COPY – Is used to copy options and tables from a previous release of FDRPAS, FDRMOVE, and FDRERASE (see next page).

AUDIT – Displays the values and descriptions of all options that are not currently set to the INNOVATION DATA PROCESSING default, that is, an audit of all changed options.

RESET – Resets all options to the default values distributed by INNOVATION DATA PROCESSING.

"SAVE" and "CANCEL" can be used on any of the option panels. Other commands are only valid on this panel. If an option value is changed since the last "SAVE" command and you attempt to exit from this panel, you are prompted to "SAVE" or "CANCEL" the changes before exiting.

New Installations

NOTE: INNOVATION DATA PROCESSING suggests that new installations should display each documented option panel and review the options on them.

Currently the only option panels used with FDRPAS, FDRMOVE, and FDRERASE are:

1 – Security Options
2 – General Options
14 – FDRPAS and FDRERASE options

Existing Installations

NOTE: The COPY command on the Global Options Panel (A.I.4) can be used to copy the option values set in a previous version of FDRPAS, FDRMOVE, and FDRERASE by reading the FDROPT module from the previous load library and setting the equivalent options in the new FDROPT.

WARNING: ***Do not copy the FDROPT module from a previous version using any standard copy utility (such as IEBCOPY or ISPF COPY); the dialog COPY function copies option values while preserving version information and new defaults in the new FDROPT.***

COPY FDR GLOBAL OPTIONS – OPTION COPY

```
---------- FDR INSTALLATION -- COPY FDR GLOBAL OPTIONS -----------------------
COMMAND ===>

WARNING: PLEASE MAKE SURE THAT THE COPY TO/FROM DATA SETS NAMES ARE CORRECT.

TO CANCEL THE COPY OPERATION, EITHER PRESS THE END KEY (PF3) OR TYPE "CANCEL".

COPY "TO"  DATA SET:    'IDP.PAS5480.LOAD'

COPY "FROM" FDR PROGRAM LIBRARY DATA SET:
  DATA SET NAME    ===> 'IDP.PAS5478.LOAD'
  VOLUME SERIAL    ===>

IN ADDITION, COPY THE FOLLOWING OPTIONS MODULES:

  ALLOCATE - ABR RESTORE ALLOCATION LIST... ===> NO    (YES NO)
  ARCPROT  - ABR ARCHIVE PROTECT LIST..... ===> NO    (YES NO)
  ABRPROT  - ABR BACKUP  PROTECT LIST..... ===> NO    (YES NO)
  RESTPROT - ABR RESTORE PROTECT LIST..... ===> NO    (YES NO)
  SCRPROT  - ABR SCRATCH PROTECT LIST..... ===> NO    (YES NO)
  CPKUNMOV - COMPAKTOR UNMOVABLE TABLE.... ===> NO    (YES NO)
  FDRNORG  - FDRREORG NOREORG LIST........ ===> NO    (YES NO)
```

Enter the data set name (and optional volume serial) of the FDRPAS, FDRMOVE, and FDRERASE program library containing the previous version. Only those options that are not set to the INNOVATION DATA PROCESSING default in effect for that version are copied; this way, if the INNOVATION DATA PROCESSING default for an option is changed in the new version, the new default is not overridden with the old default. The option values copied are immediately saved in the "to" program library; no SAVE command is required.

The options modules listed at the bottom of the screen are not used with FDRPAS or FDRERASE. The CPKUNMOV options module **IS** used with FDRMOVE.

FDRMOVE uses the COMPAKTOR Unmovable Table (CPKUNMOV) to identify data sets that should be excluded from the FDRMOVE operation. See "Unmovable Table" in Section 325.11 for the use of this table by FDRMOVE and how to update the entries in this table. If you have FDRMOVE from a previous version, you should change the CPKUNMOV value to "YES".

380.14 SECURITY OPTIONS (PANEL A.I.4.1)

FDRPAS and FDRERASE, as a default, do not invoke any type of security on individual volumes or data sets. Since they do not open individual data sets, security checks are bypassed for operations unless you enable the ALLCALL security option documented below. **By default, ALLCALL is disabled; however, FACILITY class security checks are always done.**

PANEL A.I.4.1 – SET FDR GLOBAL SECURITY OPTIONS

By default, every data set moved by FDRMOVE invokes security checks from the z/OS system allocation and catalog functions. The user id that FDRMOVE is running **MUST** be authorized to create and update **ALL** selected data sets. The security system overhead may be significant if many data sets are involved. For this reason, the default security is **not recommended** and the use of the security STGADMIN profile for FDRMOVE is recommended (see below).

SET FDR GLOBAL SECURITY OPTIONS – PANEL A.I.4.1

```
---------- FDR INSTALLATION -- SET FDR GLOBAL SECURITY OPTIONS ------------
COMMAND ===>


ALLCALL   RACF ALWAYS CALL OPTION ENABLED.............................. NO

NOABSTRK  ABSOLUTE TRACK OPERATIONS ALLOWED............................ YES

NONEW     RENAME USING NEWDD, NEWNAME, NEWINDEX AND NEWGROUP ALLOWED. YES
```

ALLCALL

If set to YES, FDRPAS does SAF-compatible security checks for volumes to be swapped. ALLCALL is set to NO (disabled) by default.

ALLCALL causes an SAF call in the form of RACROUTE REQUEST=AUTH to be used for volume-level protection. For FDRPAS, the user must have authority in class DASDVOL to the volume serial number of the online volume being swapped. For a SWAP or SWAPBULIDIX operation, ALTER authority is required, while a SWAPDUMP operation requires READ authority. If the user does not have the appropriate authority, the operation is terminated.

If the DASDVOL profile is not defined for the volume, then FDRPAS checks for the appropriate authority to every data set on the volume, in the DATASET class. This can be time-consuming and may cause swap failures if the user does not have sufficient authority, so the ALLCALL option is not recommended unless the volumes to be swapped are protected by DASDVOL profiles.

FDRERASE does not issue any DASDVOL or DATASET security calls, since the volumes it is erasing are offline and may not have valid volume serials.

FDRPAS also issues SAF calls to verify that the user has at least READ authority to a resource in the FACILITY class. The resource names are:

- **FDRPAS.SWAP** – for SWAP operations.
- **FDRPAS.SWAPDUMP** – for SWAPDUMP operations.
- **FDRPAS.SWAPBUILDIX** – for SWAPBUILDIX operations.

FDRERASE also issues SAF calls to verify that the user has at least READ authority to a resource in the FACILITY class. The resource names are:

- **FDRERASE.ERASE** – for all ERASE operations.
- **FDRERASE.ERASEALL** – for ERASE operations that specify CHECKTARGET=NO.

These FACILITY class resources allow your installation to restrict any or all FDRPAS or FDRERASE operations to certain users. These FACILITY checks are always done, even if ALLCALL is not enabled.

WARNING: ***If your installation has not protected the appropriate resource name, the operation continues and any user can execute these functions. If you do not have an active security system, SAF indicates that the resource is not protected. However, if you protect all resources by default (such as the RACF PROTECTALL option), then you need to define these resources and authorize the appropriate users for READ access.***

To reduce overhead for FDRPAS when ALLCALL is enabled, FDRPAS supports a security STGADMIN profile that allows FDRPAS to bypass security while moving DASD volumes. To invoke this support:

- Authorize the user id under which FDRPAS runs to profile STGADMIN.ADR.STGADMIN.DUMP in class FACILITY (any authority, including READ is adequate). All known security systems support such profiles.
- If the user id is authorized to that profile, all security checks from all system components invoked by the FDRPAS job are bypassed. It does not affect any other jobs.

The advantages of STGADMIN are:

- Security overhead is reduced. The user id under which FDRPAS runs is authorized to move any data set, but has no authority to those data sets outside of FDRPAS. This may be a significant advantage if a third party contractor is running FDRPAS at your installation.

RECOMMENDATION: ***INNOVATION DATA PROCESSING recommends the use of the STGADMIN.ADR.STGADMIN.DUMP profile for all FDRPAS operations when ALLCALL is enabled.***

To reduce overhead and better control security, FDRMOVE supports a security STGADMIN profile that allows FDRMOVE to bypass security while moving data sets. To invoke this support:

- Specify the STGADMIN operand on the MOVE or FASTMOVE statement, for example, `FASTMOVE TYPE=DSF,STGADMIN,`*`other operands`*.
- Authorize the user id under which FDRMOVE runs to profile STGADMIN.ADR.STGADMIN.MOVE in class FACILITY (any authority, including READ, is adequate). All known security systems support such profiles.
- If the user id is authorized to that profile, all security checks from all system components invoked by the FDRMOVE job are bypassed. It does not affect any other jobs.

The advantages of STGADMIN are:

- Security overhead is reduced. The user id under which FDRMOVE runs is authorized to move any data set, but has no authority to those data sets outside of FDRMOVE. This may be a significant advantage if a third party contractor is running FDRMOVE at your installation.

RECOMMENDATION: ***INNOVATION DATA PROCESSING strongly recommends specifying the STGADMIN operand in FDRMOVE jobs to use the STGADMIN.ADR.STGADMIN.MOVE profile for all FDRMOVE operations.***

NOABSTRK

NONEW

Not used with FDRPAS, FDRMOVE, and FDRERASE.

380.15 GENERAL OPTIONS (PANEL A.I.4.2)

PANEL A.I.4.2 – SET FDR GLOBAL GENERAL OPTIONS

These options are used by many FDR programs. A few of them are used with FDRPAS, FDRMOVE, and FDRERASE. They can be displayed or changed on the following panel.

SET FDR GLOBAL GENERAL OPTIONS – PANEL A.I.4.2

```
---------- FDR INSTALLATION -- SET FDR GLOBAL GENERAL OPTIONS -----------------
COMMAND ===>

ICFCORE    ICF VSAM CLUSTER AND COMPONENT NAMES TABLE SIZE (BYTES)... DEFAULT
FDRCC      FDR AND FDRDSF ERROR RETURN CODE.......................... ABEND
FDRSTMT    PROGRAM FDR MUST REQUIRE PARM OR SYSIN INPUT.............. NO

LINECNT    MAXIMUM NUMBER OF LINES TO BE PRINTED ON REPORTS.......... 58
SELTERR    TREAT DATA SET SELECTION FAILURE AS AN ERROR.............. YES
ALCRSTIN   INTERFACE WITH ALLOCATION CONTROL PRODUCT FOR NON-VSAM.... NO

RESTLRDT   ALWAYS RESTORE DATA SET LAST REFERENCE DATE FROM BACKUP... NO
RESTCRDT   ALWAYS RESTORE DATA SET CREATION DATE FROM BACKUP......... NO
RESTEXDT   ALWAYS RESTORE DATA SET EXPIRATION DATE FROM BACKUP....... NO

HFSQUIESCE QUIESCE HFS DURING BACKUP.............................. NO

ROUTECODE    WTO ROUTING CODES ......( 2,11                              )
DESCRIPTCODE WTO DESCRIPTOR CODES ...( 2                                 )
```

FDRCC

If an error occurs during execution of FDRPAS, FDRMOVE, or FDRERASE that is not severe enough to cause immediate termination, the program completes processing and then issues an ABEND or sets a return code to call attention to the error. The default "ABEND" causes an ABEND U0888. If you wish a return code instead of the ABEND, set any desired value from 8 to 255.

SELTERR

FDRERASE produces a diagnostic message if an offline DASD device was specified but it was not eligible to be erased. If set to "YES", FDRERASE considers this a possible user error and causes a return code or U0888 ABEND at step end to draw attention to it. If you do not want to consider this an error condition, change SELTERR to "NO" (the diagnostic message is still printed but it is not considered an error). This option can be overridden at execution time.

ROUTECODE

DESCRIPTCODE

The Write-To-Operator (WTO) routing and descriptor codes used for most of the operator messages in FDRPAS, FDRMOVE, and FDRERASE can be changed. Multiple codes can be specified for each. The routing and descriptor codes are documented in the IBM manual *z/OS MVS Routing and Descriptor Codes (SA22-7624)* and are used by z/OS to control on which consoles the messages appear, and how they are displayed.

The default routing codes of "2" (operator information) and "11" (programmer information) normally ensure that the messages appear on a system console and also in the JOBLOG of the job. If you do not want the operator to see these messages, you can change the routing codes to just "11". However, depending on system console definitions, routing code "11" WTO messages may still appear on a system console.

ICFCORE

FDRSTMT

ALCRSTIN

RESTLRDT

RESTCRDT

RESTEXDT

HFSQUIESCE

Not used with FDRPAS, FDRMOVE, and FDRERASE.

PANEL A.I.4.5 – SET FDR GLOBAL DATA SET NAME OPTIONS

This panel controls naming conventions and data set names used by ABR. Quotes are not required around any of the names specified on this panel; they are assumed to be fully-qualified names.

SET FDR GLOBAL DATA SET NAME OPTIONS – PANEL A.I.4.5

```
---------- FDR INSTALLATION -- SET FDR GLOBAL DATA SET NAME OPTIONS -----------
COMMAND ===>

ABRINDEX  HIGH LEVEL INDEX FOR BACKUPS AND ABR MODEL DSCBS........... FDRABR.
SCRINDEX  HIGH LEVEL INDEX FOR DATA SETS IN THE ABR SCRATCH CATALOG.. #.

DSNCK     ARCHIVE AND REMOTE QUEUE DATA SET NAMES ARE STANDARD....... YES

LASTAPE   LAST TAPE OPTION DATA SET NAME PREFIX............ FDRABR.LASTAPE

POOLDISK  POOLDISK  OPTION DATA SET NAME PREFIX............ FDRABR.POOLDISK

ARCDSN    ARCHIVE CONTROL FILE.... FDRABR.ARCHIVE

          REMOTE QUEUE DATA SETS
ARCHDMPQ  ARCHIVE DUMP QUEUE...... FDRABR.ABRARDQ.DATA
ARCHRSTQ  ARCHIVE RESTORE QUEUE... FDRABR.ABRARCH.DATA
BKUPDMPQ  BACKUP DUMP QUEUE....... FDRABR.ABRBKDQ.DATA
BKUPRSTQ  BACKUP RESTORE QUEUE.... FDRABR.ABRREST.DATA
```

ABRINDEX

ABR normally uses a first level index of 'FDRABR' for all backup and ARCHIVE data sets created by FDRABR (Chapters 50 and 51 describe the ABR naming conventions). In addition, this index level is used in the names of the ABR Model DSCBs that are placed in the VTOCs of every volume initialized for ABR processing "FDRABR.V*volser*". It is highly recommended that the standard index name of "FDRABR" be used but if you must use a different name change it here; ABRINDEX must be a single index (ABR adds a period at the end if you omit it).

NOTE: If you have changed the value of the ABRINDEX option, you need to change that before running FDRMOVE to ensure that ABR Model DSCBs are excluded from the move process.

SCRINDEX

DSNCHK

LASTAPE

POOLDISK

ARCDSN

ARCHDMPQ

ARCHRSTQ

BKUPDMPQ

BKUPRSTQ

Not used with FDRPAS, FDRMOVE, and FDRERASE.

380.16 FDRPAS AND FDRERASE OPTIONS (PANEL A.I.4.14)

PANEL A.I.4.14 – SET FDRPAS & FDRERASE OPTIONS

SET FDRPAS & FDRERASE OPTIONS – PANEL A.I.4.14

```
---------- FDR Installation -- Set FDRPAS & FDRERASE Options  -----------------
COMMAND ===>

            FDRPAS PLUG & SWAP Options

PASPROC     Name of cataloged procedure used for PLUG & SWAP.......... PASPROC

PASINDEX    High level index for FDRPAS catalog entries............... FDRPAS

NONRESPONDING FDRW68 message 'FDRPAS monitor failed to respond' reply DEFAULT

            ( DEFAULT allow all replies  FAIL no replies  RETRY only )


            FDRERASE Options

CHKTARGNO   Allow CHECKTARGET=NO keyword in FDRERASE control statement YES
```

PASPROC

FDRPAS requires that you install a cataloged procedure (PROC) for FDRPAS in an appropriate JES procedure library. This PROC is used when an FDRPAS MONITOR task has to dynamically invoke another MONITOR task to handle the swap of a specific volume. It can also be used in user-created FDRPAS batch jobs and to start FDRPAS started procedures from the console with START (S) commands.

The default procedure name is PASPROC. If you install this procedure with a different name, you must change it here.

The model for this procedure is found in the FDRPAS Installation Control Library (ICL) under member name PASPROC. It looks like:

```
//PASPROC  PROC PROG=FDRPAS,
//             LIB=library,      <= FDRPAS PROGRAM LIBRARY
//             EMAIL=NULLFILE,   EMAIL STATEMENT INPUT DATA SET
//             IN=NULLFILE,      FDRPAS STATEMENT INPUT DATA SET
//             OUT='SYSOUT=X',   FDRPAS SYSOUT
//             D=SHR             INPUT DATA SET DISPOSITION
//PAS      EXEC PGM=&PROG,REGION=0M
//*************************************************************
//*  FDR PLUG AND SWAP                                        *
//*************************************************************
//STEPLIB  DD DISP=SHR,DSN=&LIB
//SYSPRINT DD &OUT
//FDRSUMM  DD &OUT
//SYSUDUMP DD &OUT
//FDREMAIL DD DSN=&EMAIL,DISP=SHR
//SYSIN    DD DSN=&IN,DISP=&D
```

You must change the value of the LIB parameter to the FDRPAS load library you specified during installation. If SYSOUT class X is not a held class in your system, you may want to change it to a held class.

SECURITY WARNING:

If you have restricted access to the FDRPAS program library or enabled the FDRPAS ALLCALL security option, you may need to assign an appropriate security user id to started tasks that use PASPROC. Consult the documentation for your security product for details.

PASINDEX

FDRPAS uses the PASINDEX value as the high-level index of data set names it catalogs to create history records for FDRPAS SWAP operations. Only catalog entries are created with this name; FDRPAS does not create any real data sets using this high-level index. This name should be defined in the master catalog of every system on which FDRPAS runs, as an alias of a user catalog into which these FDRPAS catalog entries are placed. If possible, this user catalog should be a shared catalog accessible to every system, but if this is not possible, it can point to different user catalogs on different systems.

The FDRPAS SWAP and MONITOR tasks must have authority to catalog data sets beginning with the PASINDEX into the aliased user catalog. If they do not, the history records are not created but the swaps run successfully.

Do not assign PASINDEX to a user catalog with a lot of activity. Every SWAP and MONITOR task attempts to catalog into that catalog when a swap ends, there are catalog updates from multiple systems in a very short period of time. Depending on the type of catalog sharing and caching in use, this can cause other catalog requests to be delayed and may cause FDRPAS to be delayed. If desired, you can DEFINE a new user catalog for the PASINDEX.

You may change PASINDEX to any valid high-level index.

Here is an example of the IDCAMS input necessary to assign FDRPAS (or whatever value you assign to PASINDEX) as an alias of an existing ICF catalog (the IDCAMS job must be authorized to update the master catalog):

```
DEFINE   ALIAS(NAME(FDRPAS) RELATE(CATALOG.MISC))
```

If you do not assign a catalog alias for the PASINDEX, a diagnostic message FDR247 is generated when FDRPAS tries to catalog the history record, although this is not considered an error and does not cause a non-zero return code. However, if you wish to avoid the message, change PASINDEX to "#BYPASS" (customers outside the US: replace the # character with whatever character generates hex value X'7B').

NONRESPONDING

This option controls whether replies are allowed to be specified at the console for the FDRW68 message "FDRPAS MONITOR FAILED TO RESPOND". The valid values are:

DEFAULT – Allows all replies.

FAIL – Allows no replies.

RETRY – Allows replies of "**RETRY**" or "**NO**".

CHKTARGNO

This option controls whether CHECKTARGET=NO is accepted on an FDRERASE main statement.

NO – Does not accept CHECKTARGET=NO that allows you to erase volumes that are not empty or that have a valid volume label (such as the source DASD of a successful FDRPAS swap). This can be used to ensure that volumes containing data that you need are not inadvertently erased. If you specify the CHECKTARGET=NO option, it results in message FDR302 REASON=I when CHKTARGNO is set to NO.

YES – Accepts CHECKTARGET=NO that allows you to erase any volume, even though it contains data sets and valid volume label.

Default: YES.

NOTE: If you set CHKTARGNO to NO, and you plan to use FDRERASE to erase your data from DR DASD after a Disaster Recovery test (or a real disaster), you need to reset the option to YES and specify CHECKTARGET=NO to do so, since those DASD volumes will not be empty.

380.17 AUTHORIZING PROGRAMS

AUTHORIZING THE FDRPAS / FDRMOVE / FDRERASE PROGRAM LIBRARY

FDRPAS, FDRMOVE, and FDRERASE must execute as an APF-authorized program on all systems involved (both SWAP tasks and MONITOR tasks must run as authorized tasks).

If your installation has a dynamic authorized program library list (APF list), then you can authorize the program library temporarily (until the next IPL) by using the console command:

```
SETPROG APF,ADD,DSNAME=fdrpas.loadlib,VOL=volser
```

If your APF list is not dynamic, then you must update the PROG*xx* member of PARMLIB with the library name and volume serial number, and issue the console command:

```
SET PROG=xx
```

If you expect to continue to use FDRPAS, FDRMOVE, or FDRERASE beyond the next IPL, then you should update the PROG*xx* member even if you have used SETPROG to authorize it.

AUTHORIZING THE FDRPAS ISPF PROGRAM

Before the FDRPAS ISPF dialogs can be used on a given system, you must add program FDRPASA to the list of TSO authorized programs on that system and you may need to update CA ACF2 if you are using that security system.

TSO programs are authorized by modifying member IKJTSO*xx* in SYS1.PARMLIB. Program name FDRPASA must be added to both the AUTHPGM and AUTHTSF lists in that member. If you have the proper authority, you may issue the TSO command:

```
PARMLIB UPDATE(xx)
```

to activate the updated IKJTSO*xx* member immediately; otherwise it is activated after the next IPL. Once it has been activated, you can use the FDRPAS ISPF interface.

CA ACF2 COMMAND LIMITING FACILITY

If you are using the CA ACF2 Command Limiting Facility to limit the use of TSO command processors, you must add FDRPASA and FDRPASIS to the list of authorized commands for any user who uses the FDRPAS ISPF dialogs.

380.18 DYNAMIC EXIT INSTALLATION PROCEDURE

In many installations, 90% of the data sets are inactive (not in use) or become inactive over some period of time (such as when batch jobs that are using the data sets finish or when TSO users logoff). These data sets typically remain inactive for a long period of time. FDRPAS and FDRMOVE typically move these data sets without disruption. However, some data sets may be needed during the move process.

During the move process, FDRPAS and FDRMOVE hold an exclusive SYSDSN enqueue on each data set while it is actively being moved, so the data set would not be available to other applications or jobs. There can be a small number of data sets that would attempt to be accessed by an application or job after the move process has started and before the move has completed, which would usually only occur during a move of a large data set using normal I/O. The FDRPAS/FDRMOVE Dynamic Allocation exit addresses the applications that attempt to access these data sets during the move process. With the exit installed, dynamic allocations are monitored and checked to see if the data set is actively being moved by FDRPAS or FDRMOVE. If it is, this dynamic allocation is intercepted and delayed. This delay is comparable to a recall of an archived data set by a task that requires a data set that was archived.

DYNAMIC EXIT INSTALLATION

The following section describes the procedure for the dynamic installation of the FDRPAS/FDRMOVE exit.

The exit, FDR0009I, intercepts all dynamic allocations looking for data sets that are being held by FDRPAS and FDRMOVE. All dynamic allocations (SVC 99) are intercepted and the return codes checked. Successful dynamic allocations are returned. Failed dynamic allocations are checked for the return code indicating “data set in use”. If FDRMOVE is holding the enqueue, the dynamic allocation is delayed for a time value (default is two seconds) and the dynamic allocation retried. This process is repeated for a specified number of times (default is 255 times). The default wait time using these defaults would be eight minutes. After the wait time has expired (default of eight minutes), the dynamic allocation is returned to the user for processing.

The exit, FDR0009I, is dynamically installed by program FDRMVDAX. The JCL to install FDR0009I for production (to intercept all dynamic allocations for all FDRPAS and FDRMOVE related jobs) while accepting the coded defaults is as follows.

```
//FDRMVDAX JOB 'INSTALL THE FDRPAS/FDRMOVE EXIT'
//FDRMVDAX EXEC PGM=FDRMVDAX
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSLIB   DD DISP=SHR,DSN=fdrpas.loadlib
//SYSUDUMP DD SYSOUT=*
```

This JCL is supplied in member **FDRMVDAX** in the FDR Installation Control Library (ICL).

Program FDRMVDAX supports the following keywords that are specified on the PARM field:

OPERANDS

PROD

TEST

Specifies how the exit is to be used.

PROD – Is used to intercept ALL dynamic allocations.

TEST – Is used to only intercept dynamic allocations from selected jobs.

DEBUG

Display startup values as well as trace data for the dynamic allocations that are intercepted.

JOBNAME=

***jobname* –** Specifies the job(s) that are to be processed by the TEST version of the exit. JOBNAME= may be abbreviated as JOB=. This keyword can only be specified with the TEST keyword. The value specified may be:

- A single job name, for example, JOBNAME=XYZ
- A job name prefix followed by an asterisk, for example, JOBNAME=XYZ*
- Up to five job names or prefixes enclosed in parenthesis, for example, JOB=(ABC,XYZ*,SYS*).

TSO user ids and started task names are also checked for a match and are eligible for processing by the TEST version of the exit. Any job that does not match is not processed by the exit. If JOBNAME=* or JOB=* is specified, then all jobs are processed by the TEST version of the exit; in effect, this would be a simulation of the PROD keyword.

LOOPS=

***nnn* –** Maximum retry loop count. Valid values are from 1 to 800.

Default: 255.

WAITIME=

***n* –** Wait time in seconds between loops. Valid values are from 1 to 9.

Default: 2.

INSTALL THE FDRMOVE DYNAMIC ALLOCATION EXIT

To install the FDRPAS/FDRMOVE Dynamic Allocation exit and give FDR0009I a maximum of 40 minutes to hold the data sets that have been intercepted as being moved by an FDRPAS or FDRMOVE job (with the recommended production values):

```
//FDRMVDAX JOB 'INSTALL THE FDRPAS/FDRMOVE EXIT'
//FDRMVDAX EXEC PGM=FDRMVDAX,PARM='WAITIME=3,LOOPS=800'
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSLIB   DD DISP=SHR,DSN=fdrpas.loadlib
//SYSUDUMP DD SYSOUT=*
```

The Dynamic Allocation exit can be removed or deactivated by using program FDRMVDAD.

To deactivate a production install of the Dynamic Allocation exit, use the following JCL.

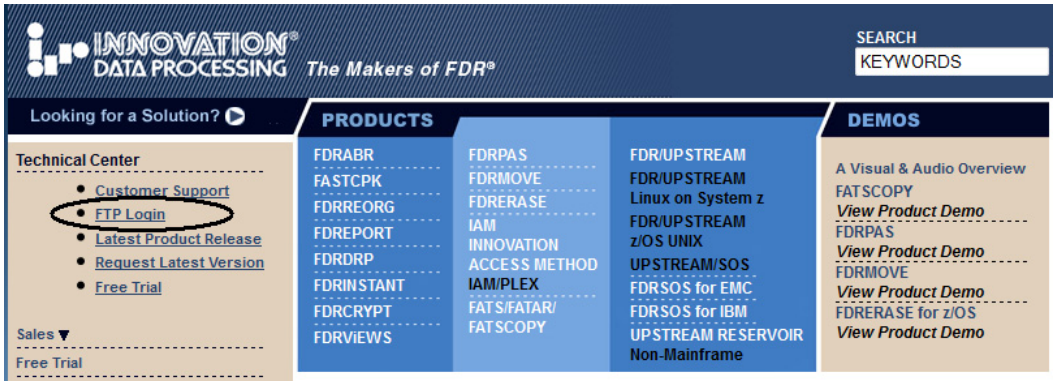
```
//FDRMVDAD JOB 'DEACTIVATE THE FDRPAS/FDRMOVE EXIT'
//FDRMVDAD EXEC PGM=FDRMVDAD
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSLIB   DD DISP=SHR,DSN=fdrpas.loadlib
//SYSUDUMP DD SYSOUT=*
```

This JCL is supplied in member FDRMVDAD in the FDR Installation Control Library (ICL).

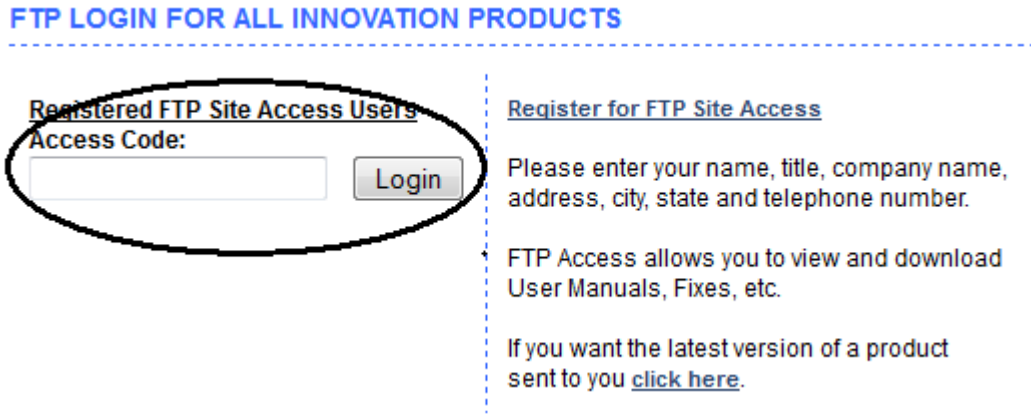
NOTE: If a version of the exit is currently installed and an FDRPAS or FDRMOVE job is actively moving data sets that may cause jobs to be intercepted by the exit, you should not replace or deactivate the exit until the FDRPAS and FDRMOVE job(s) have completed or been successfully suspended.

380.19 MAINTENANCE PROCEDURES

Fixes to the FDRPAS products are usually provided by INNOVATION DATA PROCESSING in the form of zaps. Before using FDRPAS, you should check to see if there is any new FDRPAS maintenance required since the FDRPAS installation package was created. FDRPAS maintenance is available on the INNOVATION DATA PROCESSING FTP site accessible from the INNOVATION DATA PROCESSING homepage: http://www.fdr.com



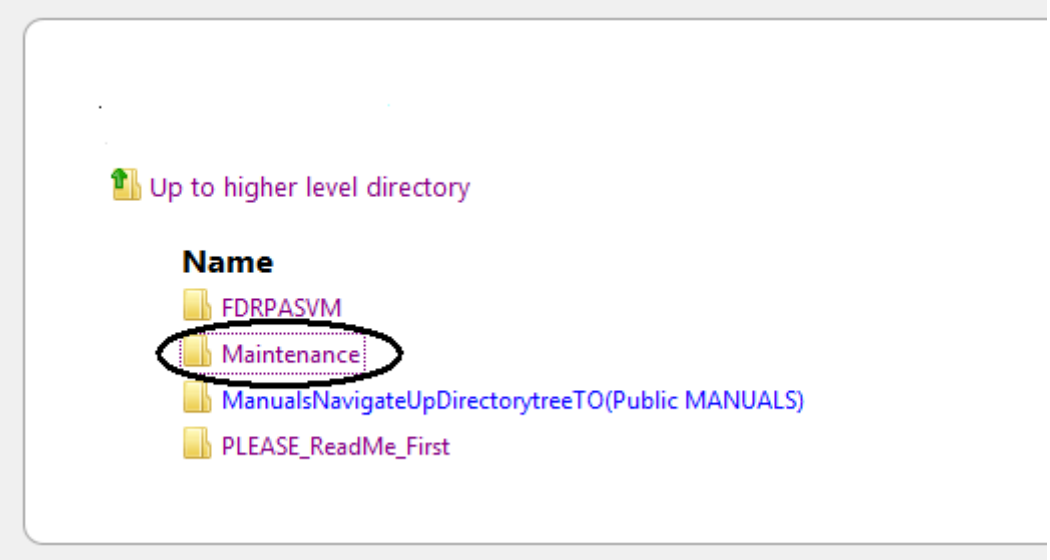
On the FTP Login screen, type your unique FTP Site Access code and click the Login button If you do not have an Access Code, please click the “Register for FTP Site Access” link shown to the right of the login area below


FTP LOGIN FOR ALL INNOVATION PRODUCTS
Registered FTP Site Access Users
Access Code:
Login
Register for FTP Site Access
Please enter your name, title, company name, address, city, state and telephone number.
FTP Access allows you to view and download User Manuals, Fixes, etc.
If you want the latest version of a product sent to you click here.

The access codes can either be 8 or 10 digits.

- 8 digit codes are for products other than FDRPAS.
- 10 digit codes are for FDRPAS products

FDRPAS maintenance is stored in the "Maintenance" directory on the FTP site that is presented to you when logging in. Click this directory to see the maintenance files.



The contents of the "Maintenance" directory has various files that are needed depending on the software product level you are at. A description of the necessary files are in the "Table_of_Fixes" PDF file that is available for the version of FDRPAS you are running. Select and open the appropriate Table_of_Fixes file.

The Table_of_Fixes PDF document looks similar to this and has multiple pages describing the necessary maintenance as well as the information on how to apply the maintenance. Please review this entire document.

FDRPAS/FDRMOVE 5.4/80 Table of Fixes

FDRPAS Product Level	Created	PTFs Applied
5.4/80	10/14/2013	N/A

Unapplied PTFs
None.

Applying FDRPAS/FDRMOVE PTFs

Fixes to the FDR products are usually provided by INNOVATION DATA PROCESSING in the form of zaps. These zaps can be applied using the IBM SUPERZAP program (PGM=AMASPZAP or IMASPZAP) but INNOVATION DATA PROCESSING also provides a program for applying zaps that has several advantages.

FDRSPZAP

The FDRSPZAP program uses JCL almost identical to the IBM SUPERZAP program and accepts all SUPERZAP control statements. However, it has an advantage over SUPERZAP:

- The CHECKSUM statement, which INNOVATION DATA PROCESSING provides as part of all zaps, is used as a check that zaps have been entered correctly. It compares the value provided to a checksum of all hexadecimal digits on the preceding VER and REP statements; almost all typing errors result in a checksum mismatch and an error message. However, the IBM SUPERZAP program compares the checksum AFTER it has already applied the changes in the REP statements. Therefore, it tells you that you have just applied a bad zap and leaves it to you to back it off. FDRSPZAP reads all the control statements and verifies the CHECKSUM before invoking SUPERZAP to perform the actual zap.

FDRSPZAP JCL

The JCL for execution of FDRSPZAP looks like this (similar to SUPERZAP):

```
//FDRSPZAP EXEC PGM=FDRSPZAP,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=  FDRPAS/FDRMOVE loadlib
//SYSLIB   DD DISP=SHR,DSN=  FDRPAS/FDRMOVE loadlib
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  .SUPERZAP control statements supplied by INNOVATION.
/*
```

These zaps can be applied by using the IBM SUPERZAP program (PGM=AMASPZAP or IMASPZAP) but INNOVATION DATA PROCESSING also provides a program for applying zaps that has several advantages.

FDRSPZAP PROGRAM

The FDRSPZAP program uses JCL almost identical to the IBM SUPERZAP program and accepts all SUPERZAP control statements. However, it has two advantages over SUPERZAP:

1. The CHECKSUM statement, which INNOVATION DATA PROCESSING provides as part of all zaps, is used as a check that zaps have been entered correctly. It compares the value provides to a checksum of all hexadecimal digits on the preceding VER and REP statements; almost all typing errors result in a checksum mismatch and an error message. However, the IBM SUPERZAP program compares the checksum AFTER it has already applied the changes in the REP statements. Therefore, it tells you that you have just applied a bad zap and leaves it to you to back it off. FDRSPZAP reads all the control statements and verifies the CHECKSUM before invoking SUPERZAP to perform the actual zap.
2. FDRSPZAP can apply zaps to the in-storage copies of the FDR system exits. If test exits are installed, fixes can be applied to either the test or production exits.

FDRSPZAP JCL

The JCL for execution of FDRSPZAP looks like this (similar to SUPERZAP):

```
//FDRSPZAP EXEC PGM=FDRSPZAP,REGION=0M,PARM=parm
//STEPLIB  DD DISP=SHR,DSN=fdrpas.program.library
//SYSLIB   DD DISP=SHR,DSN=fdrpas.program.library
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  .SUPERZAP control stmts supplied by INNOVATION DATA PROCESSING.
/*
```

FDRSPZAP PARAMETERS (PARM=)

FDRSPZAP accepts a PARM= value ("*parm*" in the example JCL above). Possible values are:

null or PARM= omitted

FDRSPZAP applies the zap only to the library indicated by SYSLIB.

IGNIDRFULL

All INNOVATION DATA PROCESSING zaps include an IDRDATA statement, which stores a string with each module indicating which zaps are applied. If the space reserved for IDR data in a load module becomes full, IGNIDRFULL allows the zap to be applied anyway (although the IDRDATA is not saved).

THIS PAGE INTENTIONALLY LEFT BLANK

390 MESSAGES AND CODES

390.1 INTRODUCTION TO MESSAGES

This chapter describes the various printer and console messages that may be output by FDRPAS, FDRMOVE, and FDRERASE and the various ABEND codes with which the programs may terminate.

FDR MESSAGE FORMAT

Messages from all components of FDR have this format:

FDR*nnn* – *message-text*

Where "*nnn*" is a 3-digit message number. When "FDR*nnn*" is followed by "**" (2 asterisks) the message usually indicates some sort of error, which may result in an ABEND or a non-zero return code. Messages without the asterisks are usually informational.

Messages directed specifically to the z/OS console or TSO user have this format:

FDRW*nn* – *message-text*

Some of these messages require replies by the z/OS operator or TSO user. However, other "FDR*nnn*" messages may also be sent to the z/OS console.

Many FDRPAS messages contain a "*system ID*" or "*sysid*", used to identify systems in a multi-system environment. The system ID displayed is taken from field CVTSNAME in the Communication Vector Table (CVT) of each system.

RETURN CODES

FDR programs set a return code at the end of the step, unless they ABEND:

00 – Normal completion

32 – A trial version of an FDR product has reached the end of its trial period and no longer functions. If you have licensed a production version of the product, you should install it. Contact INNOVATION Technical Support for assistance.

Any other return code – Errors of some kind have occurred during this execution. Check the listing for the error messages. This usually indicates that errors occurred that were not severe enough to terminate the operation, yet the program wanted to call attention to the error messages at the end of processing. Severe errors usually result in an immediate ABEND.

ABEND CODES

FDR user ABEND codes range from U0100 to U0999 and are documented in Section 390.4 "ABEND Codes". Most user ABENDs are preceded by an FDR error message.

Note that user ABEND U0888 is a special case. It is issued by FDR programs to indicate that errors occurred that were not severe enough to terminate the FDR operation, yet FDR wanted to call attention to the error messages at the end of processing. Severe errors usually result in an immediate ABEND.

Some messages might be documented only in the *FDR User Manual*. See Section 100 in the *FDR User Manual* for messages that are not documented in this *FDRPAS User Manual*.

MINI-DUMP

In many cases, an error message is followed by a set of diagnostic displays, called an FDR mini-dump and it is identified by messages beginning with FDR09*x*. The mini-dump includes:

- The general registers at the time of the mini-dump. These may or may not be significant, depending on the error.
- Blocks of storage identified by the FDR program requesting the mini-dump. These usually have a title above them for identification. The storage displayed varies depending on the error message.
- For certain I/O errors, it displays the DCB, UCB, and IOB. The IOB contains:

Bytes 2-3 – Sense bytes 0-1.

Byte 4 – I/O termination post code (7F = normal termination).

Bytes 8-15 – Hardware-generated CSW (channel status word, see the IBM *z/Architecture Principles of Operation (SA22-7832)* manual), consisting of:

Bytes 8-11 – Address of last CCW executed +8.
Bytes 12-13 – Channel status flags.
Bytes 14-15 – remaining length in last CCW (CCW length field minus this gives bytes transferred).

- It also formats the CCW chain executed. CCW(-0) identifies the last CCW executed.

I/O ERRORS AND TRACES

For most I/O errors, FDR formats information about the error in a condensed format that includes:

- The IOB (I/O request block) as described above.
- The CCW chain (four CCWs per line in most cases).
- Up to eight bytes of data associated with each CCW (16 or 20 bytes for certain CCWs). The lines containing the data alternate with the lines containing the CCWs, so that the data for each CCW is immediately below the CCW itself.

INNOVATION Technical Support may request that you add operands to your control statements that trace all I/Os issued by FDR. This same format is used for those traces.

390.2 CONSOLE MESSAGES

The following write-to-operator (WTO) and write-to-operator-with-reply (WTOR) are issued by FDR:

FDRW01 CONFIRM REQUEST TO *function* VOL=*vvvvvv* TO UNIT=*uuuu* ON *nnn* SYSTEMS REPLY YES OR NO

Reason: FDRPAS has been requested to process volume "*vvvvvv*" to offline target device "*uuuu*" and CONFMESS=YES was specified. "*nnn*" system images are participating in the function. "*function*" is SWAP for a SWAP function or DUMP for a SWAPDUMP function. If you are using the FDRPAS ISPF panels to monitor active swaps, this message also appears on the ISPF display; you can reply to it from ISPF or from the operator console.

NOTE: If WTOR=NO is specified on the FDRPAS main statement, then the **FDRW01** message is issued as a WTO, not a WTOR, and the operator cannot reply; the FDRPAS ISPF panel must be used to reply.

Action: Reply **YES** – Operator approves request.

Reply **NO** – Operator does not approve the request; the function is terminated. Message FDR234 with REASON=Q is printed.

To perform SWAP or SWAPDUMP without operator intervention, specify CONFMESS=NO on the SWAP statement or omit CONFMESS= (**NO** is the default).

CONFIRM REQUEST TO ERASE *nnnn* UNITS - REPLY YES OR NO - SEE PREVIOUS FDR235 MSG FOR UNITS

Reason: This form of message **FDRW01** is issued by FDRERASE if CONFERASE=YES is specified. It is issued once for all the units to be erased (the units are listed in message FDR235 that precedes the **FDRW01**).

Action: Reply **YES** – Operator approves request.

Reply **NO** – Operator does not approve the request; the function is terminated. Message FDR234 REASON=Q is printed.

To perform FDRERASE without operator intervention, specify CONFERASE=NO on the SWAP statement or omit CONFERASE= (NO is the default).

CONFIRM REQUEST TO INITV *nnnn* UNITS - REPLY YES OR NO - SEE PREVIOUS FDR235 MSG FOR UNITS

Reason: This form of message **FDRW01** is issued by FDRINITV if CONFERASE=YES is specified. It is issued once for all the units to be initialized (the units are listed in message FDR235 that precedes the **FDRW01**).

Action: Reply **YES** – Operator approves request.

Reply **NO** – Operator does not approve the request; the function is terminated. Message FDR234 REASON=Q is printed.

To perform INITV without operator intervention, specify CONFERASE=NO on the SWAP statement or omit CONFERASE= (NO is the default).

CONFIRM REQUEST TO VARY OFFLINE FOR *function* VOL=*vvvvvv* UNIT=*uuuu* - REPLY YES, NO OR ALL

Reason: This form of message **FDRW01** is issued by FDRERASE or FDRINITV if ONLINE=VARYOFF is specified. It is issued for each online unit that is varied offline and erased or initialized.

Action: Reply **YES** – Operator approves request for this volume.

Reply **NO** – Operator does not approve the request; the erase of this volume is terminated. Message FDR234 REASON=Q is printed.

Reply **ALL** – Operator approves request for this volume and all subsequent volumes in this step. No further **FDRW01** messages are issued.

FDRW07 JOB=*jobname* READY TO CONFIRM *function* REPLY YES TO CONFIRM

Reason: A `MONITOR TYPE=CONFIRMSWAP` or `TYPE=CONFIRMSPLIT` was submitted with `CONFMESS=YES` specified. When all the volumes in the CONFIRM step are synchronized, this **FDRW07** message is issued.

Action: Reply "**YES"** when you are ready to confirm the volumes and complete the SWAP or SWAPDUMP operation on those volumes.

`CONFMESS=YES` can be useful with `TYPE=CONFIRM`*`xxxx`* when you use a system automation product. The product can wait for the **FDRW07** message, perform other functions such as quiescing some applications, and then confirm the swaps by replying "**YES**".

FDRW60 FDROPSPF VER *x.x*/*xx* - PROCESSING ERROR - PARM-LVL *level*

Reason: The FDR installation dialog CLIST invoked program FDROPSPF with an incompatible parameter level, indicating that the DDNAME allocations for the FDR CLIST library and the FDR program library point to libraries at different maintenance levels. On a re-installation, the problem might be that the new FDR program library is not allocated to either ISPLLIB or STEPLIB DDNAMEs, thus causing the prior level FDR programs to be loaded from the Linklist.

Action: Use the ABRALLOC CLIST supplied by INNOVATION Technical Support, as documented in Section 380.5 "Invoking the Installation ISPF Dialog (Panel A.I)".

FDRW45 MIM ACTIVE -- MOVES ON OTHER SYSTEMS WILL NOT BE DETECTED

Reason: FDRMOVE detected that MIM is the cross-system enqueue product. This is a warning to notify the site that data set moves on other systems are not detected by FDRMOVE.

Action: The MIM cross-system enqueue product does not support GQSCAN, so FDRMOVE can not determine if a job on another system moves a data set. To reduce the window, where this is a factor, use FASTMOVE if possible or ensure that data sets are not being moved by other jobs.

FDRW66 *function* OF VOL=*vvvvvv* TO UNIT=*uuuu* STARTED ON *nnn* SYSTEMS (*sysid1* ...)

Reason: FDRPAS has begun to process volume *vvvvvv* to offline unit *uuuu*. *nnn* systems (1-128) systems are participating in the function. The system names of the participating systems are listed (if all system names cannot fit in one message, additional **FDRW66** messages with only the additional system names are issued). "*function*" is SWAP for a SWAP function or DUMP for a SWAPDUMP function.

function OF VOL=*vvvvvv* TO UNIT=*uuuu* CONTINUES WITH OVERRIDE OF WARNINGS

Reason: This form of the **FDRW66** message occurs when the swap has been allowed to continue despite warning messages. For example, you replied YE" to the **FDRW68** message, to document that you have approved continuing with the swap despite the warning that an apparent insufficient number of systems are participating, or you specified `LARGERSIZE=OK` to allow a swap to a larger DASD. "*function*" is SWAP for a SWAP function or DUMP for a SWAPDUMP function.

SWAP OF VOL=*vvvvvv* TO UNIT=*uuuu* NEEDS TO BE STARTED ON *nnn* SYSTEMS

Reason: This form of the **FDRW66** message occurs for a SIMSWAP operation and appears only in the printout, not on the console. It indicates that FDRPAS has determined that the volume "*vvvvvv*" is attached to "*nnn*" systems (not including systems for which `EXCLUDE` statements were present) and a MONITOR task or SWAP task should be running on each of those systems. If FDRPAS was unable to determine the number of systems, "*nnn*" is the value you specified for `#SYSTEMS=`*`nnn`*.

FDRW68 CAUTION REQUEST TO SWAP VOL=*vvvvvv* TO UNIT=*uuuu* ON *nnn* SYSTEMS IGNORING *mmm* NON-RESPONDING CPUS REPLY YES, NO OR RETRY

Reason: FDRPAS has been requested to process volume "*vvvvvv*" to offline target device "*uuuu*". "*nnn*" system images have indicated that they will participate in the function within a time limit imposed by FDRPAS, but FDRPAS expected that a larger number of systems would participate. This message may occur:

- If you have systems that are not running z/OS-type operating systems or that are not active. See "Multi-System Determination" in Section 320.1 for details.
- If you have not started an FDRPAS MONITOR task on all system images.
- If the target device is online to some systems (see message FDR235 in the MONITOR task listings).
- If the MONITOR tasks are not monitoring the proper target device(s).
- If some MONITOR tasks have failed.
- If the MONITOR tasks have a low dispatching priority on a busy system, or the LPAR running the MONITOR task has a low priority; you probably need to increase the task or LPAR priority to get a successful swap.

Message FDR234 with REASON=M may also be displayed on the console to identify the non-responding systems. See "System Determination" in Section 300.5 for an explanation of FDRPAS multi-system operation and the determination of the number of systems. See "#SYSTEMS=" in Section 310.2. If you are using the FDRPAS ISPF panels to monitor active swaps, this message also appears on the ISPF display; you can reply to it from ISPF or from the operator console.

NOTE: If WTOR=NO is specified on the FDRPAS main statement, then the FDRW68 message is issued as a WTO, not a WTOR, and the operator cannot reply; the FDRPAS ISPF panel must be used to reply. If you specify `NONRESPONDING=FAIL`, then no message is issued and FDRPAS acts as if "NO" was replied (terminate the swap).

Action: Reply **YES** – if you want the swap to continue. This should be done only if you are sure that the additional system images do not have the source volume online; INNOVATION Technical Support suggests that you use console commands on every system to verify how many have the volume online. If in doubt, reply "**NO**" until the condition can be investigated and corrected if necessary. You should always contact INNOVATION Technical Support before responding "**YES**" to this message.

Reply **NO** – terminate the swap request. Message FDR234 REASON=Q is printed.

Reply **RETRY** – FDRPAS waits additional time to see if the additional systems indicate their participation. If you did not start FDRPAS MONITOR tasks on all systems, start them before replying. If sufficient additional systems do not participate, the message is reissued.

WARNING: ***Under no circumstances should you automate the reply to this message. Every occurrence of the message must be investigated by a knowledgeable person before replying. INNOVATION Technical Support recommends that you reply "RETRY" at least once in case some MONITOR tasks were running but were not able to respond in time.***

NOTE: If you are running FDRPAS with FDRMOVE or specified `NONRESPONING=RETRY` in FDRPAS, then the FDRW68 reply only accepts "**NO**" or "**RETRY**".

FDRW89 FDR - TRIAL VERSION FROM INNOVATION DATA PROCESSING EXPIRES IN *nnn* DAYS

Reason: This is a trial version of FDRPAS/FDRMOVE or FDRERASE. The number of days the trial remains active is displayed. When there are 10 or fewer days before the trial is due to expire, this message becomes non-deletable.

Action: When the trial expires, it is not usable. If you have licensed a production version of FDRPAS/FDRMOVE or FDRERASE, you should install it in place of the trial version. Call INNOVATION Technical Support if you need assistance. This message can also come out on a rental version when the rental period is about to or has expired.

390.3 MESSAGES

CPK320I COMPAKTOR OPTIONS IN EFFECT -- CHECKSOURCE----------0X*nnnn*

Reason: Issued to inform you that the `CHECKSOURCE` option is in effect.

NOTE: This option is not valid for VM volumes.

CPK425 UNABLE TO PROCESS VOL=*vvvvvv* - DUMMY VTOC OR CMS VOLUME

Reason: The **`CHECKSOURCE=YES`** operand was specified for a volume with a DUMMY VTOC or a CMS volume.

Action: Do not use the **`CHECKSOURCE=YES`** operand when processing z/VM, Linux, or CMS volumes.

CPK600I COMPAKTOR *status* CHECKSOURCE RUN ------ VOLUME=0X*nnnn*

Reason: Informs you of the status of the run. "*status*" is **SUCCESSFULLY COMPLETED** or COMPLETED WITH ERRORS; in the latter case, some warning messages were issued.

FDR001 *program*/*function* -- VER *v.r*/*mmt* -- INNOVATION DATA PROCESSING DATE=*yyyy.ddd* PAGE *nnn*

Reason: This is the FDR page heading, containing the name of program or FDR function generating the message and the version level of FDR. "*v.r*" indicates the version and release (for example, 5.4), "*mm*" is a 2-digit number indicating the maintenance level and "*t*" is "**P**" for a production version or "**T**" for a trial.

FDR003 NONSTANDARD RECORD ZERO -- *cccchhhhrrkkllll* -- *function* CONTINUING

Reason: FDRPAS detected a nonstandard record zero (R0) in a track on the volume. The count field of the R0 is printed in hex. A standard IBM record zero format has an "*rrkkllll*" field of 00000008 – that is, record number zero, key length zero, and data length eight.

Action: FDRPAS continues copying data, to detect additional errors, but does not complete the swap. If necessary, contact INNOVATION Technical Support for assistance in correcting the error.

FDR007 *startend* TIME OF *function* -- *hh.mm.ss* - UNIT=*disktype*, IN=*inputdd* ,OUTPUT=*outdd*

Reason: Documents the time that FDRPAS, FDRMOVE, or FDRERASE began or ended an operation. "*startend*" is STARTING or ENDING. "*function*" indicates the type of function (for example, FULL VOL SWAP). "*inputdd*" is the input DD statement name, "*outdd*" is the output DD statement name (always TAPE1) and "*disktype*" identifies the input or output DASD device type (in the case of emulated DASD, such as RAMAC, the emulated DASD type, such as 3390, is shown.

FDR008 OPEN ERROR OR NO DD STATEMENT DD=*ddname* - *function* BYPASSED

Reason: One of the following:

- A required DASD or tape DD statement specified by "*ddname*" was missing.
- DDname SYSPRIN*x* is missing.
- An error occurred while opening the specified "*ddname*".

Action: The DASD in error is bypassed. See the job log for possible IBM OPEN error messages.

FDR019 RACF FACILITY PROTECTION FOR BYPASS FAILED FOR *resource*

Reason: FDRPAS and FDRERASE always issue a RACROUTE call to check for READ authority to resources in the FACILITY class, which you may use to control what users can do specific operations. The user was not authorized to the proper resource. The "*resources*" are:

FACILITY class "*resource*"	Usage
FDRPAS.SWAP	All SWAP operations
FDRPAS.SWAPDUMP	All SWAPDUMP operations
FDRPAS.SWAPBUILDIX	All SWAPBUILDIX operations
FDRERASE.ERASE	All FDRERASE operations except SIMERASE
FDRERASE.ERASEALL	If CHECKTARGET=NO is specified
FDRERASE.ONLINE.VARYOFF	If ONLINE=VARYOFF is specified

If these resources have not been defined, the error may occur because all resources are protected by default (such as the RACF PROTECTALL option); in that case define the desired resources and authorize the appropriate users for READ access.

Action: A control card error is issued. Authorize the user to the resource or run FDRPAS under a different user id.

RACF FACILITY PROTECTION STGADMIN FAILED FOR *resource*

Reason: In this form of the **FDR019** message, the operand STGADMIN was specified on a MOVE or FASTMOVE statement for simplified storage administrator security authorization in FDRMOVE. The user must be authorized to the RACF resource `STGADMIN.ADR.STGADMIN.MOVE` in class FACILITY or the equivalent in other security systems, but the security check failed.

Action: A control card error is issued and the operation terminates. If the user cannot be authorized to this resource, remove the STGADMIN operand so that normal FDR security checking is done (see "Security" in Section 325.11).

FDR020 RACF VOLUME PROTECTION FAILED ON VOL=*vvvvvv*

Reason: Security checking was enabled in the FDR Global Options Table (the ALLCALL option). A security call for class DASDVOL and volume "*vvvvvv*" failed. The user does not have sufficient authority to perform this swap or erase.

Action: An U0801 ABEND is issued.

FDR021 RACF DATASET PROTECTION FAILED ON DSN=*dsname*

Reason: Security checking was enabled in the FDR Global Options Table (the ALLCALL option). A security call for class DASDVOL and volume "*vvvvvv*" indicated that the volume was not protected by DASDVOL, so FDRPAS began doing checks in class DATASET for all data sets on the volume. The user does not have sufficient authority to the data set named, so the FDRPAS operation was terminated.

Action: An U0801 ABEND is issued.

FDR023 DUMMY VTOC READ -- ENTIRE PACK WILL BE DUMPED

Reason: FDRPAS found the VTOC on this volume started and ended on cylinder zero head zero. This is the format used on volumes initialized by z/VM; that dummy VTOC is not valid.

Action: FDRPAS copies all tracks on the volume.

LINUX VOLUME VOL=*vvvvvv*

Reason: FDRPAS found that the VTOC on this volume is in the format of a Linux volume in Compatible Disk Layout (CDL). This VTOC has a DSCB describing each Linux partition on the volume as though it was a z/OS data set.

Action: FDRPAS copies all tracks that are allocated to the Linux partitions. If the physical volume is a full-pack minidisk, it is copied correctly. If the physical volume contains more than one minidisk, then only the first minidisk is copied unless a SELECT command is included to copy the entire physical volume. See Section 320.3 "FDRPAS z/VM and Linux for System z Considerations" for further information.

FDR024 INVALID EXTENT DESCRIPTOR [REASON=*reason* DSN=*dsname*]

Reason: FDR read a DSCB (Format 1 or 3) that contained an invalid extent description. If a mini-dump is printed, the first five bytes of the DSCB printed after the registers in the mini-dump are the cylinder, head, and record number (CCHHR) of the DSCB in error. If REASON= is printed, it is one of the following:

1 – ENDING CYLINDER TOO LARGE
2 – STARTING TRACK TOO LARGE
3 – ENDING CCHH BEFORE BEGIN
4 – ENDING TRACK TOO LARGE

NOTE: Reasons 2 and 4 indicate that the starting or ending track number of an extent is higher than the highest track on a cylinder, that is, higher than 14 on 3380 or 3390; not that the start or end of an extent is after the end of the pack.

If you are swapping to a smaller DASD, reason "1" probably means that a data set on the source DASD has extents beyond the end of the target DASD. For a SWAPBUILDIX, it may mean that the communication DASD specified by SWAPUNIT= is smaller than the DASD whose VTOCIX is to be rebuilt.

Action: FDRPAS terminates the swap. You must correct the VTOC error or delete the identified data set before attempting to swap the volume again. If necessary, contact INNOVATION Technical Support for assistance.

FDR032 NO VALID TAPE(X) DD (OR MOUNT) STATEMENTS WERE FOUND

Reason: FDRMOVE did not select any volumes to process from the supplied `SELECT` statement. This is usually due to a misspelling of the volume serial number(s) on the `SELECT` statement.

Action: Check the spelling of the volume serial number(s) on the `SELECT` statement and resubmit.

FDR102 PROGRAM NOT APF AUTHORIZED

Reason: Most FDR programs must be executed as an APF-authorized program in order to execute correctly. This program detected that it was not executing authorized. This most often occurs when you are testing a new version of FDR with a STEPLIB that has not been authorized.

Action: Authorize the library from which you executing FDRPAS or FDRERASE. You can update a library list in SYS1.PARMLIB and you may be able to input a console command that authorizes the library. Contact INNOVATION Technical Support if you need assistance.

FDR107 *function* SUCCESSFULLY COMPLETED VOL=*vvvvvv*

Reason: The indicated function was successfully completed on the DASD volume "*vvvvvv*".

FDR122 OPERATION STATISTICS FOR *type* VOLUME...*volser*

Reason: This is the header for a table of statistics about the FDRPAS copy operation or FDRERASE operation that was performed on the volume indicated. These statistics include all tracks copied and re-copied in all passes of Phase 3, so the byte and track counts may be higher than the actual number of tracks or bytes copied. For FDRERASE, some fields are not meaningful and are zero.

Statistics Field	Description
CYLINDERS ON VOLUME	Total number of data cylinders on volume being processed.
DATASETS PROCESSED	Number of data sets on the volume.
BYTES READ FROM DASD	Total number of bytes actually copied up from the volume.
DASD TRACKS SWAPPED/ERASED/VERIFIED	Number of data tracks copied, erased, or verified.
UPDATED TRACKS RECOPIED	Number of data tracks re-copied due to updates (FDRPAS only).
NUMBER OF ERASE PASSES	Number of erase passes on each track (FDRERASE only).
DASD EXCPS	Number of I/O requests issued to the source volume (FDRPAS) or erased volume (FDRERASE).
TARGET DASD EXCPS	Number of write I/O requests issued to the target device.
CPU TIME (SECONDS)	The CPU (TCB) time required to process this volume, in seconds and thousandths of a second.
ELAPSED TIME (MINUTES)	The actual time, in minutes and tenths, required to process this volume.
SWAP/ERASE TIME	The actual time, in minutes and tenths, required to swap or erase this volume (excluding initialization).

FDR124 FORMAT 4 ERROR REASON=*reason - id*

Reason: FDRPAS detected a problem with the Format 4 DSCB or label track on a source volume. The Format 4 DSCB describes the VTOC itself. The VTOC may be improperly formatted. "*reason*" is a reason code plus explanatory text; "*id*" is the DSCB count field. "*reason*" may be:

reason	Description
1 - MORE THAN 1 FORMAT 4	More than one Format 4 DSCB was found, or the first DSCB in the VTOC was not a Format 4. This may be due to an IBM problem.
2 - COMPAKTOR FAILURE ON VOL	COMPAKTOR was executed against the volume but it did not complete.
3 - LABEL TRACK IS INVALID	The label track (cylinder 0 track 0) does not contain a properly formatted volume label.
4 - VOL SIZE < DEVICE SIZE	On the source volume, the number of cylinders in the VTOC is less than the actual number of cylinders on the device. (This code can also be caused by certain abnormal conditions on the target device.)
6 - TOO MANY VTOC ENTRIES	ABR found too many VTOC entries on this volume and an internal table was exceeded. An U0103 ABEND follows. Contact INNOVATION Technical Support for an override.
8	An unknown DSCB type was found.
9 - VOL SIZE 1 CYL IN VTOC	The Format 4 DSCB indicates that the volume contains only one cylinder, but the physical size of the volume is more than one cylinder.

Action: FDRPAS terminates the swap. You must correct the error before you attempt to re-execute the swap. List the VTOC in hex with:

```
//LIST  EXEC PGM=IEHLIST
//SYSPRINT DD SYSOUT=*
//DISK1    DD UNIT=3390,VOL=SER=vvvvvv,DISP=OLD
  LISTVTOC VOL=3390=vvvvvv,DUMP
/*
```

Contact INNOVATION Technical Support for assistance. For reason 4 or 9, you may be able to run ICKDSF with the `REFORMAT REFVTOC` command to correct the source volume before retrying the swap.

FDR125 I/O ERROR READING THE VTOC X'*ccchhhh*' VOL=*vvvvvv* ENTIRE PACK WILL BE DUMPED

Reason: FDRPAS detected an I/O error reading the VTOC on the cylinder and track indicated in hex. An I/O trace message is also printed to document the error.

Action: Since FDRPAS cannot determine the data sets on the volume, all tracks on the source volume are copied to the target device. However, if the I/O error repeats when FDRPAS is trying to copy the VTOC track to the target device, the swap is terminated.

FDR126 VTOC/DSCB ERROR REASON=*reason*

Reason: While analyzing the Data Set Control Blocks (DSCBs) in the VTOC, FDRPAS found an invalid DSCB or another VTOC-related error occurred. "*reason*" describes the error.

Action: For most errors, FDRPAS continues processing the volume, in order to detect additional errors. However, the swap is not performed. Certain errors cause immediate termination.

FDR128 INVALID RECORD ZERO ON TRACK X'*ccchhhh*' DATA X'*ccchhhh*'

Reason: The track specified has an invalid record zero (R0). The count field of R0 should always contain the ID of the track it resides on. The cylinder and track (in hex) of the failing track is displayed, along with the count field of R0.

Action: FDRPAS continues copying data, to detect additional errors, but does not complete the swap. If necessary, contact INNOVATION Technical Support for assistance in correcting the error.

FDR129 I/O ERROR ON DISK PACK - LAST HOME/SEEK ADDRESS READ X'*ccchhhh*' [*additional text*]

Reason: An I/O error was detected by FDRPAS or FDRERASE on this source volume or target device. An IBM IOS000I message may also have been printed on the job log. For FDRERASE, this *additional text* appears on the right:

WILL CONTINUE – The erase operation continues, but one or more tracks are not erased in the identified cylinder.

WILL TERMINATE – More than 20 errors were encountered, so FDRERASE terminates this DASD volume.

Action: FDR149 I/O trace messages are printed to detail the error. FDRPAS terminates the swap on the first I/O error. FDRERASE tolerates up to 20 DASD write errors before terminating the erase.

FDR130 CYL=*ccccccc* HEAD *xxx...x* WAS COPIED

Reason: Internal message showing each cylinder and track (head) copied; it appears only if `PRINT=ALL` is specified on the SWAP statement. An "**X**" appears for each track copied within the cylinder. "*ccccccc*" is the cylinder number in decimal.

FDR149 ANTRQST *function-call* TRACE - RTNCD=*xxxx*, RSNCD=*xxx* -- …

Reason: Internal API request has failed. Return Code and Reason Code are displayed.

API FORCED: *xxxxxxxx xxxxxx*0F *RCxxxxxx xxxxxxxx xxxxxxxx*

Reason: Re-attempt is performed and Sense Message Format (x'0F') and Reason Code (x'*RC*') is displayed.

EXCP FAILED - FUNCTION *function* CCW: *xxxxxxxx xxxxxxxx*

Reason: I/O command has issued an error return code. IOB Sense information is displayed for diagnostic purposes

EXCP REQUEST - CCW: *xxxxxxxx xxxxxxxx*

Reason:

FLASHCOPY PRESERVE MIRROR - *type* [FAIL]

Reason: Informational message indicating **REMOTEINSTANT** parameter was specified to utilize Remote Mirror (Preserve Mirror) FlashCopy feature. "*type*" is **PREFERRED** or **REQUIRED**.

If the optional word "**FAIL**" is appended to the message, this means that Remote Mirror (Preserve Mirror) specified with the **REQUIRED** feature failed. FDRCOPY COPY or MOVE proceeds with normal I/O. This message is accompanied by **FDR149** IOB SENSE message(s) with Remote Mirror sense information.

INFO: FLASHCOPY TGT IS AN ACTIVE SRC

Reason:

IOB SENSE: *xxxxxxxx xxxxxx*0F *RCxxxxxx xxxxxxxx xxxxxxxx*

Reason: I/O error detected. Sense Message Format (x'0F') and Reason Code (x'*RC*') is displayed. Common Reason Code (x'*RC*') values are:

X'85' – An ESTABLISH could not be completed because the target is a PPRC primary or secondary volume, or a Global Mirror for z/Series (XRC) primary volume.

X'A7' – An ESTABLISH could not be completed because the designated source track is already an existing target track'

X'AE' – An ESTABLISH could not be completed because the designated target track is already an existing target track.

X'AF' – An ESTABLISH could not be completed because the designated target track is already an existing source track.

NORMAL I/O PERFORMED - FLASH NOT ALLOWED

Reason: FLASH operation is not allowed due to reason displayed in IOB Sense information. Copy / Move is performed using Normal I/O. Message is informational.

RETINFO ANTF0*xxxx error-description*

Reason: Provides IBM description for error condition.

tracedata

Reason: Debugging and diagnostic trace information is displayed during error handling or if "debugging" is enabled.

FDR152 ICF VSAM ERROR REASON=*rc* DSN=*cluster*/*component* VOL=*vvvvvv*

Reason: An I/O error or logical error exists within the SYS1.VVDS data set or in the VVDS information for the cluster or component indicated, on volume "*vvvvvv*" (this error may also occur for non-VSAM data sets on SMS-managed volumes). The reason code "*rc*" may be:

rc	Reason
1	VSAM clusters exist on the volume, but FDR failed to find the SYS1.VVDS data set on the output DASD or on the backup file. The VVDS may not exist or FDR encountered an error trying to find it. The IOB details the error.
2	The number of extents in the SYS1.VVDS data set was zero.
3	The SYS1.VVDS contains more than 3 extents. FDR had an I/O error or logical error finding an Format 3 DSCB. The IOB details the error.
4	An I/O error or logical error was detected when FDR read or wrote entries within the SYS1.VVDS data set, or the VVDS contains no active records (even though there are VSAM clusters on the volume). For an I/O error, the IOB details the error.
5	The length of a VVR/NVR entry within the VVDS is zero or negative.
6	The length of all of the VVR/NVR entries within a block of the VVDS exceeded the length in use within the block.
7	A VVDS record indicates that the length of its in-use data exceeds 4096.
8	An error occurred when DSF attempted to find one of the associated components for this cluster. The component may be in the VVDS but not in the VTOC. The IOB details the error.
9	FDR is unable to store all of the VSAM component and cluster names in its internal table. A large number of VSAM clusters must exist on the volume. Specify the keyword `ICFCORE=` on the `DUMP TYPE=` statement or permanently increase the ICFCORE value in the FDR Global Options Table.
A	A cluster specifies a catalog name that was not found in the catalog entries stored in the first record of the VVDS.
B	FDR could not find the volume information/allocation cell (type 23) in the VVR record, or an individual cell had an invalid length.
C	More than one SYS1.VVDS data set was found on the volume. The operating system uses the VVDS that contains the current DASD volume serial number in the name. More than one VVDS can be created if a volume with a VVDS is renamed and a VSAM cluster is allocated using the volume with the new serial number.
D	The volume serial in the data set name of the VVDS (SYS1.VVDS.V*volser*) does not match the volume serial of the DASD. This can occur if a volume was copied or restored to a new DASD volume serial but the VVDS was not renamed. Another possible cause is the `DUMPCONDITIONING` option of DFSMSdss. This is only a warning; FDRPAS swaps the volume properly, using that VVDS. However, VSAM and SMS-managed data sets on the volume are not usable until the condition is corrected.
E	FDR did a GETMAIN for above the line storage for a VSAM table but the GETMAIN failed. Try increasing the REGION= parameter to more than 32M.
F	FDR did not find both a VVR/NVR and a DSCB for a given data set; one was missing. Also occurs when FDR was unable to calculate the free space in a cluster.
G	The imbedded index VVR (Type Q) was not found.
H	The size of an extent was not a multiple of the CA size; or the high RBA of an extent exceeded the maximum. On a logical restore, the original cluster had an imbedded index (the IMBED attribute) and the high-level index component had more than one extent.
I	The control interval size (CISIZE) on the backup cluster does not match the cluster on DASD. The displacement is X'16' (see Note on page 390-15).
J	The number of CIs per CA on the backup cluster does not match the cluster on DASD. The displacement is X'0E' (see Note on page 390-15).

rc	Reason
K	The space allocation unit (TRACK vs. CYL) on the backup cluster does not match the cluster on DASD. The displacement is X'10' (See Note on page 390-15). This error is also issued following an **FDR160** message; see **FDR160** in the FDR documentation.
L	The physical block size or number of blocks per track on the backup cluster does not match the cluster on DASD. The displacement is X'11' or X'15' (see Note on page 390-15).
M	The number of tracks per CA on the backup cluster does not match the cluster on DASD (Check the secondary allocation value specified). The displacement is X'17' (See Note on page 390-15).
N	The VSAM cluster is being restored to a different type of cluster, for example, KSDS to an ESDS. Flags at two displacements are checked (see Note on page 390-15). At displacement X'03' it checks: ❖ X'02' - Relative Record (RRDS) ❖ X'04' - Key Range ❖ X'10' - Replication (REPLICATE) ❖ X'20' - Imbeded Index (IMBED) ❖ X'80' - Key Sequence (KSDS) ❖ At displacement X'2A' it checks: At displacement X'2A' it checks: ❖ X'01 - Variable RRDS (VRRDS) ❖ X'04' - Linear (LDS) If none of these flags are on, the cluster is an ESDS (entry sequenced) cluster. There may be other flags on in those bytes; they do not matter. However, all of the flags named above must match (on or off) between the input and output cluster.
O	The length of the VVR/NVR exceeds the length of the cells within the VVR/NVR.
P	The VVR entry found is in orphan and not related to a catalog entry.
Q	In a multi-level alias environment, the rename of the temporary data set name failed. Contact INNOVATION Technical Support for assistance.
R	SMS Attributes of the backup cluster and the cluster on DASD do not match. FDR checks for Extended Format (EF) and for "over 4GB addressing". The displacement is either X'03' or X'1D'.
S	The backup cluster contains a key range while the DASD does not or the key lengths are not the same. The displacement is X'24' (See Note on page 390-15).
T	A VVR entry in the VVDS contains a cell type other than a 21, 23, or 60.
U	The type of VVR entry on the backup does not match the DASD (Z record being restored to Q). The user may be attempting to restore a multi-volume component to the wrong sequence component.
V	The total length of a VVR/NVR record does not match the sum of the lengths of the cells within it, on the output DASD.
W	The total length of a VVR/NVR record does not match the sum of the lengths of the cells within it, on the backup file.
X	The number of extents specified in the VVR record does not match the DSCB entry in the VTOC for a component.
Y	A base cluster is being restored to an alternate index or vice versa.
Z	Internal Error – The length of the FDR backup file control block for VSAM was less than the sum of the lengths of the VVRs/NVRs within it.

Action: Except for REASON=D, FDRPAS terminates the swap because of the logical errors on the volume. You must investigate and fix the problem before rerunning the SWAP.

NOTE: For the indicated reason codes, FDR prints a mini-dump displaying the VVR (VSAM Volume Record) from the backup file and from the cluster on DASD. In the registers at the top of the dump, register 14 points to a cell within the DASD VVR and register 15 points to a cell within the backup VVR. Those registers plus the hex displacements shown above for the appropriate reason codes point to the fields that did not compare.

FDR153 VSAM SCRATCH ERROR COMP=*comp* CODE=*code* CLUSTER=*cluster*
VSAM SCRATCH/UPDATE ERROR COMP=*comp* CODE=*code* CLUSTER=*cluster*

Reason: FDR attempted to scratch a VSAM cluster but the CATALOG DELETE SVC failed. "*comp*" and "*code*" match the return code and reason code documented for IBM message IDC3009I except that "*comp*" values over 9000 are internal FDR errors.

Some commonly occurring IBM codes are:

comp	*code*	Reason
0008	0042	The cluster to be scratched was not cataloged. FDR can only scratch cataloged clusters.
0076	0000	User attempted to scratch a multi-volume cluster that was not aliased to a user catalog. This data set may be partially deleted.
0076	0008	User attempted to scratch a cluster with an alternate index that has the NOUPGRADE attribute on another volume.
0084	0000	The VSAM cluster is date protected. Specify VEXPD=NONE on the DUMP Statement to scratch this cluster.

comp values over 9000 are from FDR:

comp	Reason
9004	Dynamic allocation failed for the specified cluster. "*code*" is the decimal equivalent of the dynamic allocation error code; it also appears in hex in register 0 within the mini-dump. Dynamic allocation error codes are documented in the ISPF online tutorial appendix and in various IBM manuals. In particular, a dynamic allocation error code of decimal 5896 or X'1708' indicates that the input cluster was not found in the catalog.
9008	User attempted to move a multi-volume VSAM cluster. The components on this volume were copied to the output volume, but were not scratched. If all of the associated components have been dumped or copied, the user can use SUPERSCRATCH or IDCAMS to scratch the cluster.
9012	Internal Error. The TIOT search failed to find the DASD DD statement name to which the cluster was dynamically allocated.
9020	A KSDS with alternate indexes has been archived but it was not scratched because ALTINDEX=NO was specified.
9024	Cluster cannot be scratched because it is cataloged to a different volume or because an alternate index with the NOUPGRADE attribute is on a volume by itself.
9028	Cluster cannot be scratched because it is cataloged as non-VSAM or cataloged to more than 255 volumes.

Action: Check the error code and correct if necessary. An FDR mini-dump is printed displaying the CATALOG DELETE parameter list and the catalog name. If needed, contact INNOVATION Technical Support for assistance.

FDR156 ALLOCATE FAILED FOR *nnnnn quan* COMP=X'*comp*-*code*' VOL=*vvvvvv* DSN=*dsname*

ALLOCATE FAILED FOR CATALOG COMP=*comp* CODE=*code* VOL=*vvvvvv* DSN=*dsname*

Reason: FDR attempted to allocate a non-VSAM data set "*dsname*" on volume "*vvvvvv*" and the attempt failed (if FAILED FOR CATALOG is displayed, the allocation was successful, but a later attempt to catalog the data set failed; this can also occur for VSAM). "*nnnnn*" is the size of the data set that failed allocation in tracks or cylinders, "*quan*" is TRK or CYL. If the allocation was attempted on multiple volumes (because of the NVOL= operand or the ABR RESTORE ALLOCATION LIST), it failed on all of them, but "*vvvvvv*" is the first volume on which it was tried, and "comp" is the error code from that volume. If the data set is SMS-managed, "*vvvvvv*" is the volume that FDR selected for the data set and may not be the same as the volume on which SMS tried to allocate it.

"*comp*" may be return codes from the IBM ALLOCATE SVC (SVC 32), from the IBM CATALOG SVC (SVC 26), from the IBM SMS interface, or internal diagnostic codes from FDR itself.

If "**FAILED FOR CATALOG**" is displayed, "*comp*" is the return code and "*code*" is the reason code from the CATALOG SVC, in decimal, as documented for the IBM message **IDC3009I**. Some of the common catalog codes shown under message FDR157 also apply to **FDR156**. Note that if "*comp*" is **51**; look up the reason "*code*" under code **50** in message IDC3009I. If "*comp*" is **300**, this is an internal error; contact INNOVATION Technical Support for assistance. If "*comp*" is 9*xxx*, contact INNOVATION Technical Support for assistance. "*code*" equal to **00040** may mean an attempt to catalog a GDG generation when the GDG base does not exist.

Otherwise, if "*comp*" is a value less than X'1000', it is from the ALLOCATE SVC, in hex. "*code*" may be an eight-digit hex IBM diagnostic code. For z/OS, see the IBM manual *z/OS DFSMSdfp Diagnosis (GY27-7618)*.

Common DADSM CREATE (ALLOCATE) codes include:

comp	*code*	Reason
0004		Duplicate data set name in VTOC; data set already exists on the volume. This may occur for multi-volume data sets if the piece currently on the volume has a different volume sequence number than the piece being restored.
0008		VTOC or VTOC index (VTOCIX) is full.
000C		I/O error in VTOC or VTOC index (VTOCIX).
0010		Requested absolute tracks not available; an unmovable data set cannot be allocated because the required tracks are not free.
0014		Requested space quantity not available; DADSM was unable to find sufficient free space (in up to 5 extents) to allocate the data set, or the specific space required for an unmovable data set was not free.
00AC		No security authorization to create data set. You must have ALTER authority to the data set under RACF or the equivalent in other security systems.
00B4		IGGPRE00 installation exit rejected the request. A local exit or DASD Control software product has disallowed allocation on the volume that FDR selected.
00C0	SMS failed the allocation request. The most common cause is a mismatch between the SMS flags in the VTOC of the output volume and that volume's status in the current SMS configuration; common reason codes are:	
	04160053	VTOC indicates INITIAL status (in process of conversion to SMS).
	04160054	VTOC indicates non SMS-managed, but the volume is in an SMS storage group.

comp	code	Reason
	04160055	VTOC indicates SMS, but the volume is not in an SMS storage group.

Action: You can use FDRCONVT (Section 70.30) to convert the volume to SMS-managed or non SMS-managed, or you can remove the volume from its storage group in the current SMS configuration. "*comp*" values greater than 1000 are from FDR, in hex, as follows:

comp	Reason
2354	The name of the data set being restored does not meet IBM's data set naming conventions. If NEWNAME=, NEWGROUP=, or NEWINDEX= was specified, the error is in the new name; otherwise it is the original name of the data set that is invalid. You can bypass this check by specifying NODSNCHK on the RESTORE statement; the data set is allocated but the catalog probably fails.
FD*xx*	The IBM subsystem interface (IEFJSREQ) failed with return code "*xx*".
FE*xx*	The IBM SMS interface failed the allocation request with return code "*xx*", (usually 08). "*reason*" is an SMS error code, in decimal. You can find the description by looking up IBM message IGD*xxxxx*, where "*xxxxx*" is the reason code.
FF04	FDR cannot allocate an ISAM file. See member **ISAM** in ICL Library.
FF08	FDR cannot allocate a non-ICF VSAM file.
FF0C	FDR cannot allocate an unmovable file with more than three extents.
FF18	On a COPY/MOVE of a data set protected by a discrete RACF profile, an attempt to issue RACDEF to define a profile for the output data set failed."*reason*" is "0000*xxyy*" where "*xx*" is the RACF return code and "*yy*" the RACF reason code ("*xx*" of 04 indicates that the profile already exists, 08 that the user is not authorized to create the profile)
FF50	GETMAIN failure during allocation. Increase the region size.
FF51	Allocation parameter list error. Contact INNOVATION Technical Support for assistance.
FF52	FDR did not select a target output volume. This can occur if the data set's original volume or the volume you specified as NVOL= is not online. Specify a new NVOL= value or update the ABR Restore Allocation List to direct the data set to an online volume.
FF53	NVR is not record type N. Contact INNOVATION Technical Support for assistance.
FF54	NVR cell type not 22. Contact INNOVATION Technical Support for assistance.
FF55	NVR type 24 cell length invalid. Contact INNOVATION Technical Support for assistance.
FF56	NVS SMS sub cell not type 24. Contact INNOVATION Technical Support for assistance.
FF57	DSCB address not passed. Contact INNOVATION Technical Support for assistance.
FF58	The selected target DASD volume was SMS-managed, but no storage class was assigned. It may be that your SMS Automatic Class Selection (ACS) routines are assigning a null storage class to this data set. To restore as SMS-managed, specify STORCLAS=; if it still fails, use BYPASSACS or update your Automatic Class Selection (ACS) routines. To restore as non SMS-managed, specify NVOL= to direct it to a non SMS-managed volume.
FF59	Volsers in CPL cannot be located, or over 10 volumes passed. Contact INNOVATION Technical Support for assistance.
FF92	Tracks in extent not divisible by CASIZE during FDRMOVE operation. This means that the extent allocated by DADSM was not the proper size to allow the move of a multi-volume component on all volumes tried by FDRMOVE. Those volumes are probably fragmented.

Action: Determine the cause and take appropriate action. It may be possible to circumvent the problem by preallocating the output data set. If needed, contact INNOVATION Technical Support for assistance.

FDR157 VSAM ALLOCATION ERROR COMP=*comp* CODE=*code* VOL=*vvvvvv* CLUSTER=*cluster*

Reason: FDR attempted to allocate a VSAM cluster by using the IBM CATALOG SVC, but the attempt failed. "*comp*" is the return code and "*code*" is the reason code from CATALOG, in decimal, or is a value generated by FDR itself. If the allocation was attempted on multiple volumes (because of the NVOL= operand), it failed on all of them, but "*vvvvvv*" is the first volume on which it was tried, and the codes are from that volume. If the cluster is SMS-managed, "*vvvvvv*" is the volume that FDR selected for the cluster and may not be the same as the volume on which SMS tried to allocate it.

Codes from the IBM CATALOG SVC can be found documented under IBM message **IDC3009I**. Common codes include:

comp	*code*	Description
0004	00010	The catalog selected by FDR to catalog this cluster does not exist (see the description of the ICFCAT= operand). If you dumped this cluster on one system and are restoring it on another, this may mean that a catalog with the name of the original cluster's catalog does not exist; you may need to specify ICFCAT=ALIAS to catalog it into the proper catalog.
0004	00120	The catalog selected by FDR to catalog this cluster is not defined in the master catalog. See 0004-0010 above.
0008	00008	Name already in the catalog. A NEWGROUP= or NEWINDEX= value may have resulted in renaming the output cluster or some of its components to the same name.
0008	00038	Name already in the catalog. Either the cluster or component name already exists, or the original cluster is still cataloged. The VRECAT operand can be used on the RESTORE statement to delete the duplicate names.
0056	00006	Insufficient security authorization to define the cluster. You must have ALTER authority to the cluster under RACF or the equivalent in other security systems.
0068	00008	The VTOC or Indexed VTOC is full.
0068	00020	Insufficient space available to define the cluster.

If "*comp*" is a value above 9000, the error is generated by FDR itself ("*comp*" values from 9052 through 9080 may occur if the cluster being restored had an imbedded index, and at the time of the backup the index component was in multiple extents):

comp	*code*	Reason
9004		The VVR entry for the data component on the backup contains an unknown cell type.
9012		FDR cannot allocate a multi-volume component.
9016		The VVR entry for the index component on the backup contains an unknown cell type.
9024		SELECT specified a component name instead of cluster name, or a component of the requested cluster was missing from the VVDS at the time of the backup, or ICF=IGNORE was specified during the dump.
9028		FDR could not find the type 23 cell in the VVR.
9032		FDR cannot allocate a VSAM cluster that contains an alternate index to a NEWNAME=. Use NEWGROUP= or NEWINDEX= instead of NEWNAME=.
9036		FDR cannot allocate a key range VSAM cluster using NEWNAME=. Use NEWGROUP= or NEWINDEX= instead of NEWNAME=.
9040		The volume serial number in the SYS1.VVDS data set does not reflect the volume it is on. If FDR were to allocate a VSAM cluster under these conditions, the operating system would create a new VVDS data set.
9044		The name of the cluster or one of its components being restored does not meet IBM's data set naming standards.

comp	code	Reason
9048		Error occurred applying the NEWINDEX= value.
9052		A LOCATE SVC failed. CODE shows the return code from LOCATE. The reason code is not shown.
9056		Unable to find the PCCB for a user catalog.
9060		Dynamic allocation failed. CODE shows the return code in register 15 from SVC 99. The dynamic allocation return code (DARC) from the SVC 99 parameter list is not shown.
9064		VSAM OPEN failed. Message **IEC161I** may appear on the job log. CODE shows the return code from VSAM OPEN, as documented under IBM message **IEC161I**. For example, code **0040** indicates a security violation.
9068		VSAM PUT failed. CODE shows the return code from VSAM PUT.
9072		AMB Extension was not found.
9076		Internal error in reconstructing the imbedded index; RBA of index extent is not correct.
9080		VSAM EOV failed. Message **IEC070I** may appear on the job log. CODE shows the return code from VSAM EOV, as documented under IBM message **IEC161I**. For example, code **0104** or **0209** indicates that the volume that you were restoring to did not have enough space to contain the data set. It should be possible to restore the cluster to another volume with more space available.
9084		Error from RACF, for a cluster with a discrete profile. FDR successfully allocated the cluster, and then tried to define a new discrete profile, using the profile of the original cluster as a MODEL. The CODE value is the RACF return code, and register 0 within the mini-dump contains the RACF reason code. A CODE of 4 indicates that the profile already exists, 8 indicates the user is not authorized to create the profile.
9088		A KSDS with an excessive number of key ranges cannot be defined.
9092		A PAGE/SWAP cluster cannot be moved.
9093		CISIZE of input cluster is invalid for logical restore.
9094		Physical block size of input cluster is invalid for logical restore.
9095		CASIZE of input cluster does not fit in a cylinder of the output DASD for logical restore. For example, a cluster allocated in cylinders on a 3390 cannot be restored to a 3380.
9096		Allocation of a multi-volume VSAM cluster failed. This can also occur for a KSDS with the IMBED option or keyranges. "*reason*" indicates the specific error:
	00004	GETMAIN/FREEMAIN error.
	00008	Volume list from LOCATE is invalid.
	00012	Non-VSAM catalog entry invalid.
	00016	Unable to obtain LOCAL lock.
	00020	Type 23/21 cell not found in VVR.
	00024	Cluster on more than 255 volumes.
	00028	Number of extents does not match Type 60 cell.
	00032	UCB not found for output DASD.
	00036	Imbedded index extents do not match.
	00040	Length error on cluster/component name.
	00044	VVR length invalid.
	00048	NEWNAME= not supported (use NEWGROUP= or NEWINDEX=).

comp	code	Reason
	00052	Catalog entry type not non-VSAM. Multi-volume VSAM clusters that have not been completely restored are cataloged as non-VSAM (the first volume serial number is ####V*x*) but the cluster name is currently cataloged as something else (usually VSAM). This may be due to a previous incomplete restore of this cluster. See "Action".
	00056	LOCATE error. This can occur if the original catalog of the cluster is not available on the system on which the restore was done. Specify `ICFCAT=ALIAS` to restore into the aliased catalog on the current system. Also, see "Action".
	00060	Cluster has alternate index.
	00064	`VRECAT` failed.
	00068	Cluster has over 123 extents on a volume.
	00072	Number of extents on volume not correct. Probably due to free space fragmentation on the target device causing DADSM to allocate the request in multiple extents.
	00075	Unlike device restore not supported.
	00080	Attempted to move a VSAM component but it is not cataloged to the input volume.
	00084	Cataloged to over 255 volumes during FDRMOVE move of a multi-volume cluster.
	00088	Catalog entry type not non-VSAM, and VRECAT was specified, but the piece of the cluster on the current source volume is guaranteed candidate space and does not warrant deleting the cataloged cluster. The parts of the cluster that contain data may have been restored already, or may be restored subsequently.
	00092	Tracks in extent not divisible by CASIZE during FDRMOVE operation. This means that the extents allocated by DADSM were not the proper size to allow the move of a multi-volume component on all volumes tried by FDRMOVE. Those volumes are probably fragmented.
	00096	Extent update LOCATE failed during FDRMOVE move of a multi-volume cluster.
	00100	Extent update UCBADDR failed during FDRMOVE move of a multi-volume cluster.
	00104	Extent update DATANAM failed during FDRMOVE move of a multi-volume cluster.
	00108	Extent update READVVR failed during FDRMOVE move of a multi-volume cluster.
	00112	Extent update UPDTVVR failed during FDRMOVE move of a multi-volume cluster.
	00116	Extent update ENDUPDT failed during FDRMOVE move of a multi-volume cluster.
	00120	Extent update, VVR type not type 60 during FDRMOVE move of a multi-volume cluster.
	00124	Cluster not cataloged or cataloged as non-VSAM during FDRMOVE move of a multi-volume cluster.
	00128	Extent update, number of extents is less than 1 during FDRMOVE move of a multi-volume cluster.
	01*xxx*	OBTAIN error ("*xxx*" is OBTAIN error code).
	02*xxx*	DYNAMIC ALLOCATION error ("*xxx*" is allocation error code).
	03*xxx*	EXTEND error ("*xxx*" is EXTEND error code).
	03129	Insufficient space for component on this volume.
9100		The selected output DASD volume was SMS-managed, but no storage class was assigned. Specify `STORCLAS=` or select a non SMS-managed volume, and resubmit.
91*xx*		The IBM SMS interface failed the allocation request with return code "*xx*" (usually 08). "*reason*" is an SMS error code, in decimal. You can find the description by looking up IBM message **IGD*xxxxx***, where "*xxxxx*" is the reason code.
92*xx*		The IBM subsystem interface (IEFJSREQ) failed with return code "*xx*".

comp	code	Reason
9450		GETMAIN failure during allocation. Increase the region size.
9451		Allocation parameter list error. Contact INNOVATION Technical Support for assistance.
9452		UCB address not passed or invalid. Occurs if the data set is not SMS-managed (no storage class assigned), but FDR has not selected a volume on which to allocate it. Specify NVOL= and rerun.
9458		BYPASSSMS was specified and the selected output DASD volume was SMS-managed, but no storage class was assigned. Specify STORCLAS= or select a non SMS-managed volume, and resubmit.
9459		Volsers in CPL cannot be located, or over 10 volumes passed. Contact INNOVATION Technical Support for assistance.

Action: An FDR mini-dump is printed displaying the CATALOG DEFINE parameter list and related data areas. Determine the cause and take appropriate action. For errors involving names already in the catalog, the VRECAT operand on the RESTORE statement may be used to DELETE the duplicate names from the catalog (but should be used cautiously since it may DELETE other clusters). If needed, contact INNOVATION Technical Support for assistance.

FDR158 DATA SET IS ACTIVE DSN=*dsname*

Reason: FDRPAS or FDRMOVE has found that the data set named is active (enqueued to another job or task on this system or another system).

Action: FDRPAS copies all the tracks allocated to the data set.

For FDRMOVE, this is normal for active data sets; FDRMOVE retries the data set until it becomes available.

FDR159 UNABLE TO MOVE *reason* VOL=*vvvvvv* DSN=*dsname*

Reason: If the "*reason*" indicates "**A CATALOG**", then a catalog was selected. FDRMOVE cannot move a catalog. "*dsname*" is the catalog or CATINDEX name.

If the "*reason*" indicates "**IN CPK RECOVERY**", then the "*dsname*" data set that is listed was involved in a prior CPK function that failed and is still in recovery state. The data set is not moved by FDRMOVE.

If the "*reason*" indicates "**VSAM ORPHAN**", a problem was found in the VSAM data set structure for the VSAM data set listed in "*dsname*". The VSAM data set entry is in the VTOC, but not in the VVDS.

If this message appears with the text "**REASON=**", consult Chapter 100 of the standard FDR manual for an explanation. This should not occur when using the standard FDRMOVE unmovable table.

FDR162 SMS ERROR MESSAGE FOR DSN=*dsname message text*

Reason: The allocation of an SMS-managed data set failed. The diagnostic and informational messages generated by SMS during this allocation are printed.

Action: Examine the messages to determine the cause of the allocation failure.

FDR164 CANDIDATE-SPACE ALLOCATION BYPASSED DSN=*dsname*

Reason: The current backup or input DASD contains an extent of a multi-volume VSAM cluster that was flagged as a "candidate space". A candidate space is created when a multi-volume SMS cluster is allocated with a storage class with the "guaranteed space" attribute and real volume serials are specified. SMS preallocates extents on the volume, but they are not part of the active extents of the cluster until it expands and uses them.

Action: The candidate space is bypassed and is not allocated. FDR does not restore VSAM candidate volumes, either regular or candidate-space.

FDR170 DEVICE IS NOT ELIGIBLE FOR *function* UNIT=*uuuu* VOLID=*volid reason* [OVERRIDDEN]

Reason: The indicated DASD device is not eligible to be erased by FDRERASE or initialized by FDRINITV for the reason indicated. If VOLID= is displayed, this is the ID in the first four bytes of the volume label (VOL1 indicates a normal volume label, FDR*n* indicates a volume label modified by FDRPAS, FDRERASE, or FDRINSTANT.

If the text **OVERRIDDEN** appears, then the condition is overridden by an operand (such as `ACTIVETARGET=PROCESS`). If VOLID is VOL1, this message may be preceded by an FDR255 message if the volume contains data sets.

Action: The DASD is not erased unless the condition is overridden.

DEVICE IS NOT ELIGIBLE FOR *function* UNIT=*uuuu* VOL=*vvvvvv* ONLINE TO CPU=*cupid*

Reason: This form of the **FDR170** message is issued when the target device is found to be online. An active pathgroup relationship for an EMC subsystem can also cause this message to be issued.

Action: This DASD is not erased unless `ACTIVETARGET=PROCESS` was specified. For an active pathgroup relationship in an EMC subsystem, you can vary the device online followed by varying the device offline to break the relationship.

DEVICE IS ALREADY ERASED UNIT=*uuuu* VOL=*vvvvvv* [OVERRIDDEN]

Reason: This form of the **FDR170** message is issued when the target device has already been successfully erased. The text OVERRIDDEN appears if the condition was overridden by `ALREADYERASED=PROCESS`.

Action: The DASD is not erased unless `ALREADYERASED=PROCESS` was specified.

DEVICE IS ELIGIBLE FOR *function* UNIT=*uuuu* VOL=*vvvvvv* VOLID=*volid* CU=*mfrseral*/*ssid*-*diskid*

Reason: This form of the **FDR170** message is issued only for SIMERASE operations in FDRERASE and FDRINITV INITVTOC operations. The indicated DASD device is eligible to be erased by FDRERASE and is erased by a real erase operation or is eligible to have a VTOC built. "*mfr*" is the DASD subsystem manufacturer (for example, IBM), "*seral*" is the subsystem serial number, "*ssid*" is the subsystem ID (SSID), and "*diskid*" is the internal DASD identification, if it can be determined, or the device address, if not.

***function* BYPASSED UNIT=*uuuu* DOES NOT HAVE A VALID VTOC AND VTOC LOCATION/VOLSER NOT PROVIDED**

Reason: This form of the **FDR170** message is issued only for FDRERASE `EMPTYVTOC` or FDRINITV `INITVTOC` operations. The indicated DASD device does not have a valid volume label and/or VTOC.

Action: The DASD is not re-initialized unless the condition is overridden. Specify the VTOC location and size on the MOUNT statement. Specify `CHANGEVOL=` if there is not a proper volume serial.

program BYPASSED UNIT=*uuuu* CHANGEVOL NUMBER WRAPPED

Reason: This form of the **FDR170** message is issued by FDRERASE or FDRINITV when volumes were being relabeled by a `CHANGEVOL=` parameter that included slashes "/" to generate a sequence number. The sequence number exceeded its maximum value for the number of slashes specified.

FDR171 ERASE RELEASED ITS CSA STORAGE

Reason: FDRERASE released the small area of CSA it uses to track active and completed erase operations.

FDR172 ERASE STARTED/ENDED PASS *n* PATTERN=*hh*

Reason: FDRERASE has started or ended pass "*n*" on a DASD volume, using pattern byte X'*hh*'. See the description of the `ERASEPASS=` and `ERASEPATTERN=` operands for details. The pattern is "ERASE" if the tracks are being hardware erased (ERASE pattern "01").

FDR173 ERASE HARDENED DATA TO UNIT=*uuuu* IN *nnn* SECS *errortext internal*

Reason: At the end of a pass for ERASE or SECUREERASE, FDRERASE issued commands to ensure that the overwriting data has been "hardened" from cache to the physical DASD. This took "*nnn*" seconds. If an error occurred while hardening the data, "*errortext*" appears; it can be "I/O ERROR" or "TIMEOUT". TIMEOUT indicates that the control unit has not acknowledged the hardening of the data for over the HARDENTIMEOUT value (default 2 minutes), or the count of remaining tracks has not decreased for some time. "*internal*" is some internal numbers about the hardening operation. Note that this message may appear in the erase subtask listing (SYSPRT*xx*) or the SWAP task listing (SYSPRINT).

Action: FDRERASE has successfully erased or overwritten the DASD. However, if "*errortext*" appears, FDRERASE cannot be certain when the data is actually written to the physical DASD, but it continues with the erase operation.

FDR174 *function* REWROTE VTOC TO UNIT=*uuuu* [*additional text*]

Reason: For FDRERASE ERASE operations and FDRINITV INITVTOC operations, CPYVOLID=YES was specified and FDRERASE or FDRINITV invoked ICKDSF to rewrite an empty VTOC on the volume (FDRERASE erases it first). "*additional text*" can be:

AND REBUILT INDEXED VTOC – A VTOCIX was also allocated and initialize on the volume.

FAILED CHECK IXSYSP*xx* – ICKDSF failed, check the IXSYSP*xx* DD statement output for error messages.

FAILED IXSYS ERROR – An error occurred dynamically allocating or opening IXSYSP*xx* or IXSYSX*xx*.

FDR175 PRINT UNIT=*uuuu* CYL.*ccccc* TRK.*tt* REC.*rrr* KL.*kl* DL.*dl* DATA=*data*

Reason: A PRINT statement was used with FDRERASE, which requests that information about the contents of tracks be printed from an offline or online DASD. One **FDR175** message is printed for each record on the selected tracks. However, if ERASEDATA=NONE was specified, only Record 0 (R0) is printed for each track, to reduce the size of the printout.

The cylinder (*ccccc*), track (*tt*), and record number (*rrr*) are displayed, along with the key length (*kl*) and data length (*dl*) of the record; all values are in decimal. "*data*" is the first and last eight bytes of the record, in hexadecimal plus the first eight bytes in EBCDIC. If all data bytes in the record are the same, the last eight hex bytes are replaced with the text "**ALL BYTES THE SAME**"; this is the expected result if you have used the ERASE or SECUREERASE function to erase a DASD unless ERASEPATTERN=FE was used (for SECUREERASE or ERASE with ERASEPASS, the results of the last pass are shown). If "**END OF FILE**" is displayed, then the printed record is an EOF (a record with a data length of 0).

Every track has a record 0 (R0), so at least one line is printed per track. If there are no other records on the track, the text "**NO RECORDS ON TRACK**" appears to the right of the eight-byte data field of R0. If there are other records on the track "***nn* RECORDS ON TRACK**" is displayed on the R0 line. After an ERASE or SECUREERASE function, you see one record on the track.

FDR177 VERIFY CHECKED UNIT=*uuuu* AND ALL TRACKS CONTAINED ERASE DATA PATTERN=*xx*

Reason: A VERIFY function was executed on DASD "*uuuu*" and all selected tracks contained data written by FDRERASE. "*xx*" is a single hexadecimal byte if all bytes in all records contained that byte. It is "**FE** (Random)" if the tracks contained the random pattern generated by ERASEPATTERN=FE. If the DASD was erased using multiple passes (such as SECUREERASE), the displayed pattern is the last one used.

VERIFY CHECKED UNIT=*uuuu* AND ALL TRACKS CONTAINED NO RECORDS PATTERN=ERASE

Reason: In this form of the **FDR177** message, a VERIFY function was executed on DASD "*uuuu*" and all selected tracks were empty (containing no records).

VERIFY CHECKED UNIT=*uuuu* AND ALL TRACKS DID NOT CONTAIN ERASE DATA

Reason: In this form of the **FDR177** message, a VERIFY function was executed on DASD "*uuuu*" and one or more selected tracks contained records that were not written by FDRERASE. This may mean that the DASD was not erased, or the erase failed.

This can also occur if the CHANGEVOL operand was specified on the MOUNT statement. The CHANGEVOL operand places a volume serial number in CYL 0 and this causes the VERIFY function to not have erase data in that location causing this message to be printed.

Action: FDR175 messages are issued for each non-erased track to indicate its contents. If more than ERASENUMB= tracks fail verification, the VERIFY function terminates on this volume. The step terminates with a non-zero return code.

FDR181 MOVE REQUEST ALREADY ON NVOL=*volser* DSN=*dsname*

Reason: An FDRMOVE request found that a selected data set was already on a volume in the NVOL= list, so the data set is not moved.

MOVE REQUEST HAS YET TO MOVE VOL=*volser* DSN=*dsname*

Reason: In this form of the **FDR181** message, the specified data set has not yet been moved.

FDR182 FDR REQUESTS THAT VOL=*volser* MOVE TO/FROM THE TRANSIT STATION

FDRPAS *status* SWAP TASK TO MOVE TO TRANSIT STATION VOL=*vvvvvv*

Reason: The first form of this message indicates a FASTMOVE job has requested that the SWAP `TYPE=TRANSIT` job move the specified volume to a transit station DASD volume, or move it back to its original device.

The second form of this message indicates that the transit job has received the move request. The "*status*" keyword indicates the current status of the move request and indicates STARTED, ENDED, or DELAYED. DELAYED indicates that is a delay is processing the request that can be due to the maximum number of transit jobs already running or all of the available target units are already being used.

FDR183 FDR SMS DISABLENEW VOL=*volser*

Reason: `DISABLENEW=YES` was specified on a MOVE or FASTMOVE statement, and the source volumes were SMS-managed, so FDRMOVE marked the source volumes as DISABLENEW to prevent new allocations.

FDR MARKED PRIVATE VOL=*volser*

Reason: In this form of the **FDR183** message, `DISABLENEW=YES` was specified on a MOVE or FASTMOVE statement, and the source volumes were non SMS-managed, so FDRMOVE changed their mount status to PRIVATE. This prevents new non-specific allocations to those volumes but specific volume allocations are still allowed.

FDR SUBMITTED FDRPAS TRANSIT JOBNAME=*jobname*

Reason: In this form of the **FDR183** message, FASTMOVE determined that one or more of the source volumes must be moved to transit stations, so it submitted the internal FDRPAS "`SWAP TYPE=TRANSIT`" specified by the PASJOB DD statement. A SIMMOVE waits up to 15 seconds for the SWAP job to start; FASTMOVE waits up to 10 minutes. If the job does not start, the FDRMOVE job fails.

FDR SUBMITTED FDRPAS MONITOR JOBNAME=*jobname* FOR CPUID=*cpuid*

Reason: In this form of the **FDR183** message, FDRPAS determined that a FDRPAS monitor was not running on an LPAR and submitted the job for that CPUID based on the JCL in the PASJOB DD statement.

FDR184 *status message*

Reason: This is the message produced by the console `STATUS` command (`F` *`movejob`*`,STATUS`). It has many different forms but briefly, it displays the status of volumes and data sets being processed by FDRMOVE. For FASTMOVE it shows which volumes are moving to transit stations. For all functions it shows the active data sets that FDRMOVE is waiting to move. Similar messages are displayed by the FDRMOVE ISPF panel. Examples and more details appear throughout Chapter 325 "FDRMOVE".

VOL=*volser status source - target*, *nnnnnnnn* DATASETS TO MOVE - HIGHEST SEQ# *nnnn* --[NOIX]

Reason: This form of the **FDR184** message identifies the source volume being processed and the current processing being done. Also shown are the source and target DASD subsystem as well as the number of data sets to be moved and the number of unique volumes required for multi-volume data sets. "**NOIX**" is appended to the message to indicate that either the source or target volume (or both) does not have an indexed VTOC and processing may be slower for this volume. The "*status*" for the volume level message can be:

AWAITING MOVE FROM – Volume is available to be processed, but not yet selected.

IN TRANSIT STATION – The source volume has been moved to the transit station and is now available to FDRINSTANT to move the individual data sets.

AWAITING TRANSIT – No transit station is currently available for use by this volume.

MOVING TO TRANSIT – The source volume is being moved to a transit volume so FDRINSTANT can be used to move the individual data sets.

VTOC IS NOT EMPTY – A SIMMOVE operand was specified with `VTOCEMPTY=CHECK` and it was found that not all data sets would be moved.

|---FOR ACTIVE DSN=*dsn nn* JOBS(*jobnames*)

Reason: This form of the **FDR184** message provides information on the individual data sets and the job(s) that an enqueue on the data set. Up to four job names are listed in the message.

FDR MOVED *nnnnnnnn* DATASETS/COMPONENTS SUCESSFULLY

MOVE STATUS *nnnnnnn* ACTIVE DATASETS *nnnn* AWAITING TRANSIT *nnnn* MOVING TO TRANSIT *nnnn* IN TRANSIT STATION

Reason: These two forms of the **FDR184** message is a summary of all the data sets by current status.

***nnnn* JOBS ACTIVE MIM-JOB(UNKNOWN ON OTHER LPARS**

Reason: This form of the **FDR184** message is displayed when MIM is determined to be in the system and the MIM job on the other LPARs cannot be identified.

MIM NOT ACTIVE MOVE TERMINATED VOL=*volser*

Reason: This form of the **FDR184** message is displayed when MIM is determined to be the cross-system enqueue product, but it is not active. The FDRMOVE job stops to prevent possible data set corruption.

TRANSIT JOBNAME=*jobname* WAS SUBMITTED BUT IS NOT ACTIVE

Reason: This form of the **FDR184** message can be a STATUS response and can be issued independently. It indicates that the `FDRPAS TYPE=TRANSIT` job with the indicated job name was submitted but has not become active. It can also be issued if the job was active but was terminated prematurely. In a SIMSWAP, this comes out once after 15 seconds if the job does not become active. In a FASTMOVE, it comes out every 15 seconds for five minutes, after which the FDRMOVE job fails.

One common cause of this is an invalid JOB statement in the FDRPAS job stream. Although FDRMOVE does some validation of the JOB statement (see the error message), it is not comprehensive and an invalid JOB may cause the job to fail. If you do not see the error, check SYSLOG for JES messages that may give you guidance.

FDR185 FDRMOVE ERROR REASON=*x*

Reason: FDRMOVE has detected a serious error and terminates. Many of the reasons relate to errors submitting the FDRPAS transit job. The reason code "*x*" may be:

1 **PASJOB DD FAILED TO OPEN –** The PASJOB DD statement was omitted or points to an improper data set.

2 **INTRDR DD FAILED TO OPEN –** The internal INTRDR DD statement allocated to submit the PASJOB had an error.

3 **PASJOB MISSING PROCESS TABLE –** An internal table is missing.

4 **PASJOB DOES NOT START WITH // –** The internal PASJOB does not have "//" in columns 1-2 of the first statement

5 **PASJOB DOES NOT START WITH JOB –** The internal PASJOB does not have the verb JOB in the first statement.

6 **PASJOB MOUNT EXCEEDS 16 CARDS –** If `VOL=&&&&&&` is used on the `MOUNT` statement, the `SWAPUNIT=` operand cannot be continued to more than15 statements (16 total).

7 **PASJOB MOUNT CONTINUE IS BLANK –** A continuation statement for a `MOUNT` statement was blank.

8 **I/O ERROR ON INTRDR DD –**

9 **I/O ERROR ON PASJOB DD –**

A **PASJOB DD IS EMPTY –** There is no JOB in the PASJOB DD statement.

B **JOBNAME MISSING OR OVER 8 BYTES –** On the first statement in PASJOB, which must be a JOB statement, the job name was missing or too long.

C **CANNOT FIND MODULE FDRDSF –** Internal error.

D **INSUFFICENT STORAGE AVAILABLE –** Unable to GETMAIN a needed table.

E **CONTROL CARD NAME MISMATCH –** Internal error.

F **ENQUEUE DATASET NAME BLANK –** Internal error.

G **PASJOB MOUNT STATEMENT MISSING –** The internal PASJOB must have a `MOUNT` statement.

H **PASJOB MULTIPLE MOUNT CARDS –** If the internal PASJOB uses `VOL=&&&&&&` on the `MOUNT` statement, there must be only one `MOUNT`.

J **NVOL LIST HAS NO ONLINE VOLUME –**

- `NVOL=` on an FDRMOVE `SELECT` statement does not point to any online volumes.
- `NEWCUID=` was specified without `NEWSTORGRP=` or `ENEWSTORGP=`.

K **PASJOB NAME ALREADY ACTIVE –** The job name specified in PASJOB is already running.

M **PASJOB TYPE=TRANSIT MISSING –** The first control statement to FDRPAS must specify `SWAP TYPE=TRANSIT`.

N **VOL= VOLUME SAME AS NVOL= VOL –** A `MOUNT` statement in the FDRPAS job specifies VOL= with a volume serial number matching an `NVOL=` in the FASTMOVE input

O **SMS TO NON-SMS OR VICE-VERSA –** FDRMOVE cannot move a non SMS-managed source volume to an SMS-managed target volume, or vice versa.

P **NVOL= NOT SAME DEVICE TYPE VOL=*volser* –** The `NVOL=` list for the indicated volume contains a device type that does not match the `VOL=`, such as a 3390 to a 3380.

Q ***vvvvvv*** **MOVED TO WRONG CNTLUNIT –** The volume indicated was swapped to a different control unit than the selected FDRMOVE target volumes reside in. This was moved by the FDRPAS "SWAP TYPE=TRANSIT" job, but you specified the wrong `SWAPUNIT=` values for the volume.

R **NO ONLINE VOLUMES MATCH VOL=*volser* –** In FDRMOVE:

- `VOL=` ending in an asterisk was specified but no matching volsers were found.
- `VOL=*` was specified. This is not valid.
- `CUID=` was specified without `STORGRP=`.

S **ONLINE VOLUMES EXCEED MAXCARDS –** `VOL=` ending in an asterisk was specified but the number of volumes found (including preceding statements) exceeded `MAXCARDS=`. Increase `MAXCARDS=`.

FDR186 DSN=*dsname* VOL=*volser nnnnnn* ELIGIBLE NVOLS *nvol nvol …*

Reason: The NVOLs eligible for an FDRMOVE of the indicated data set are displayed. "*nnnnnn*" is the number of NVOLs.

DSN= ALLDSN *storgrp* VOL=*volser nnnnnn* ELIGIBLE NVOLS *nvol nvol*

Reason: This format of the **FDR186** message indicates that all the data sets from the indicated storage group are moved to the NVOLs specified. "*nnnnnn*" is the number of NVOLs.

FDR187 CANDIDATE VOLUME *volser action* FOR DSN=*dsname*

Reason: The indicated non SMS-managed VSAM cluster has candidate volumes. If *action* is "**UPDATE DEFERRED**" then updating the candidate volume has been deferred until all active pieces of the cluster have been moved. "**UPDATED TO *volsr2***" indicates that the candidate was updated to an unused volume serial number in the NVOL list. If a catalog error occurs while updating the candidates, another form of the **FDR187** message displays the catalog error codes.

Action: If all occurrences of the **FDR187** message say "**UPDATE DEFERRED**" (and never "**UPDATED TO**") then there were not sufficient unused NVOLs and the candidate was not updated; in this case, you may need to use the IDCAMS command with the `REMOVEVOLUMES` and `ADDVOLUMES` operands to change the candidate list to valid volsers.

FDR188 ALL VOLUMES IN TRANSIT STATION

Reason: All required volumes referenced in this FDRMOVE FASTMOVE step have been moved to transit stations in the target control unit, but FASTMOVE is still waiting for data sets on those volumes to be released.

Action: Stop the application(s) using those data sets; you can display the data set names with the console STATUS command or the FDRMOVE ISPF interface (see Section 325.11 "FDRMOVE Special Considerations"). As soon as the data sets are dequeued, FASTMOVE moves them with whatever instant replication facility is available in the target control unit.

FDR189 SWAP OF ACTIVE JES VOL=*vvvvvv type* [*additional text*]

Reason: This is an information message indicating that FDRPAS has processed an active JES volume. The message displays the JES volume serial "*vvvvvv*". "*type*" details the type of JES volume and can be SPOOL, CHECKPOINT, or COUPLING.

"*additional text*" can be:

- **SUSPENDED-REPLY GO TO START COPY** - This message is issued before the copy of the JES spool or checkpoint volumes or a volume containing a sysplex coupling data set, if `MAXACTIVESWAPS=` and `CONFIRMSWAP` or `CONFIRMSPLIT` are specified. The purpose is to delay the copy of these special volumes until all of the other volumes have been synchronized and it is close to the time of completing the SWAP or SWAPDUMP tasks. If a `MONITOR TYPE=CONFIRMSWAP` or `TYPE=CONFIRMSPLIT` is run, "GO" is replied internally without user intervention.
- Other text detailing when the volume has been serialized. This message may be issued multiple time for a volume serial if multiple passes are performed on the volume.

FDR210 I/O ERROR ON TARGET UNIT=*uuuu* [*additional text*]

Reason: An I/O error occurred on the target device (documented by a preceding FDR149 message). Is it also issued to the console as a non-scrollable message.

Action: If `SWAPIOERR=NORETRY` was specified or defaulted, the swap is immediately terminated.

If `SWAPIOERR=RETRY` was specified, it has the *additional text* "**I/O WILL BE RETRIED EVERY 5 SECONDS"** and the failing I/O is retried until it is successful or until the swap is aborted. If the I/O is eventually successful, the message is issued again with the *additional text* "**RECOVERED**".

FDR230 *program* RELEASE ERROR - UNIT=*uuuu* REASON=*reason*

Reason: FDRERASE FDRINITV had an error attempting to do a "deleted space release" on an IBM RVA or Oracle StorageTek SVA. Reason codes include:

G **MESSAGE ID=*mmmm* STATUS CODE=*ssss* –** ECAM message number "*mmmm*" was issued to the RVA/SVA. The message was accepted but it returned status code "*ssss*". Contact INNOVATION Technical Support for assistance.

H **MSG=*mmmm* COMP *cccc* REASON=*rrrr* –** ECAM message number **"***mmmm***"** was issued to the RVA/SVA. The message was rejected with completion code "*cccc*" and reason code "*rrrr*". Contact INNOVATION Technical Support for assistance.

I **MSG=*mmmm* PARM *pppp* CC=*cccc* RS=*rrrr* –**

K **UNDEFINED SNAPSHOT STATUS –** Unexpected response from the RVA/SVA.

L **UNDEFINED SNAPSHOT MESSAGE COUNT –** Unexpected response from the RVA/SVA.

M **EXPECTED MSGID=*nnnn* - RECEIVED MSGID=*nnnn* –** Unexpected response from the RVA/SVA; the wrong ECAM message was received.

Action: FDRERASE reverts to using standard erase commands.

FDR232 *program* RELEASED PHYSICAL SPACE FROM UNIT=*uuuu*

Reason: An FDRINITV step is erasing tracks from a DASD in an IBM RVA or a Oracle StorageTek SVA subsystem. Instead of erasing each individual track, the tracks were released by an ECAM "deleted space release" request, which has the same effect but takes only seconds per volume. This is also done for an ERASE function where a pattern byte of "01" (erase) is specified.

FDR233 *systemid* (SERIAL# *sssssssss*) ACKNOWLEDGES THE *function* OF VOL=*vvvvvv* *variable-text*

Reason: This form of the **FDR233** message indicates that the FDRPAS task on the indicated system acknowledged the SWAP or SWAPDUMP request for volume "*vvvvvv*". "*systemid*" is the system name and "*sssssssss*" is the hardware serial number of that system as reported by the STIDP instruction. The unit addresses shown are the addresses on the system indicated (the same device may have different addresses on different systems). "*function*" is SWAP for a SWAP function or DUMP for a SWAPDUMP function. "*variable-text*" can be:

AND HAS JOINED IN THE *function* OF UNIT=*ssss* TO *tttt* – Indicates that the MONITOR task on the indicated system acknowledged the swap request for the volume "vvvvvv" and participates in the swap of the volume. This message prints in the output of the MONITOR task on each system, and the messages from every system print in the output of the SWAP or SWAPDUMP task for the volume.

AND WILL NOT JOIN BECAUSE SOURCE IS NOT ONLINE – Indicates that the MOINTOR task on the indicated system acknowledged the swap request for the volume "vvvvvv" and waits for the swap process but does not participate in the swap of the volume because the volume is not online on that system. This message prints in the output of the MONITOR task on each system and the messages from every system print in the output of the SWAP or SWAPDUMP task for the volume.

***mftr-info* –** This appears on the SWAP or DUMP task and indicates the hardware manufacturer and type of the source and target devices.

FOLLOWING LPARS NEED MONITOR
CPU WITH (SERIAL# *sssssssss*)

Reason: This form of the FDR233 message appears at the end of the SYSPRINT of a SIMSWAP job listing summarizing all the LPARs that need to have MONITOR tasks to process the DASD volumes in this job step. The "CPU WITH (SERIAL# *sssssssss*)" is repeated for every LPAR needing a MONITOR task.

Action: Ensure MONITOR tasks are created for all the LPARs referenced.

CPU WITH (SERIAL# *sssssssss*) IS ATTACHED TO VOL=*vvvvvv* [WAS EXCLUDED]

Reason: This form of the **FDR233** message indicates that FDRPAS has identified that the system with hardware serial "*sssssssss*" is attached to the source volume "*vvvvvv*". "**WAS EXCLUDED"** appears only if an EXCLUDE statement was included for that serial.

Action: If this is a SIMSWAP, verify that an FDRPAS MONITOR task is running on each of the identified systems (except for excluded systems), monitoring the target device for this swap, before you run the real SWAP operation. If some of the identified systems do not participate in the swap, you may need to exclude them (see "Multi-System Determination" in Section 320.1 for details).

CPU WITH (SERIAL# *sssssssss*) CANNOT DETERMINE # OF SYSTEMS ATTACHED TO VOL=*vvvvvv*

Reason: This form of the **FDR233** message is issued by a SIMSWAP operation. It indicates that FDRPAS cannot determine the systems attached to the specified volume, because it is in a subsystem that does not support that determination. "*sssssssss*" is the serial of the system where SIMSWAP is running.

Action: Use the `#SYSTEMS=` operand on the SWAP statement when swapping this volume; be sure to specify the number of systems correctly and ensure that a MONITOR task is running on each system.

FDR234 *function* ERROR ON VOL=*vvvvvv* - UNIT=*uuuu* REASON=*reason*

Reason: FDRPAS, FDRERASE, FDRINITV, or a related program had an error that caused the operation on the specified volume to the specified unit to fail. "*function*" is SWAP for a SWAP function, DUMP for a SWAPDUMP function, ERASE for an ERASE function, or INITVTOC for an INITVTOC function. Reason codes include:

0 **I/O INTERCEPTS ADDRESSES OR UNIT ADDR HAVE CHANGED –** The FDRPAS I/O intercept addresses in the DDT of the source volume or the device address of the source volume have changed unexpectedly. This probably indicates that some other program has also installed an I/O intercept on the source volume, with unknown results. The device address changed means that some other program swapped the volume that FDRPAS was processing.

1 **MOUNT STATEMENT NOT SPECIFIED –** There were no control statements following the main statement. `MOUNT` statements are required in order to define the source volume and target device for SWAP, the potential target volumes to be monitored for MONITOR, the volumes to be initialized for INITVTOC, etc. An exception is a `MONITOR TYPE=SWAP` statement that specifies `DYNMON=YES`.

2 **MOUNT STATEMENT NOT FOUND –** Some control statements were provided after the main statement, but none of them were `MOUNT` statements. See reason 1.

3 **SWAPUNIT=|ERASEUNIT=|INITUNIT= NOT SPECIFIED IN MOUNT STATEMENT OR MORE THAN 1 UNIT –** `SWAPUNIT=`, `ERASEUNIT=`, or `INITUNIT=` must be specified on this `MOUNT` statement. Following a `SWAP` statement, the `SWAPUNIT=` operand can only specify a single device. It may also be that you specified a blank preceding `SWAPUNIT=`, `ERASEUNIT=`, or `INITUNIT=` in the control statement.

4 **SOURCE AND TARGET UNIT ADDRESS ARE EQUAL –** The current device address of the volume specified by `VOL=` is equal to the device address specified by `SWAPUNIT=`.

5 **INCOMPATIBLE SOURCE AND TARGET UNIT DEVICE TYPES –** The device type of the source volume specified by `VOL=` is different from the device type of the target device specified by `SWAPUNIT=`. For example, one is a 3380 and the other is a 3390.

5 **MISMATCH SOURCE AND TARGET MIDAW FLAGS –** On a System z processor (or beyond), both the source and target devices must support MIDAWs (Modified IDAWs) or both must not. All IBM DASD subsystems support MIDAWs but some non-IBM subsystems may not. In order to swap, you must shut down MIDAW support with the console command: `SETIOS MIDAW=NO`.

6 **TARGET SWAPUNIT|ERASEUNIT|INITUNIT NOT FOUND –** The device address specified by `SWAPUNIT=`, `ERASEUNIT=`, or `INITUNIT=` was not found in the current I/O configuration.

7 **TARGET SWAPUNIT NOT DASD –** The device address specified by `SWAPUNIT=` is not configured as a DASD device.

7 **TARGET ERASEUNIT|INITUNIT NOT DASD OR NOT SUPPORTED –** The device address specified by `ERASEUNIT=` or `INITUNIT=` is not configured as a DASD device.

NOTE: The FDR system supports all types of DASD devices that are supported by the z/OS operating system, so please contact INNOVATION Technical Support for assistance if this message appears and the device is DASD.

8 **TARGET SWAPUNIT IS NOT OFFLINE ON THIS LPAR –** The device address specified by `SWAPUNIT=` is not offline on this system. If this is the intended target device for the swap, vary it offline on all systems and resubmit the swap request.

8 **TARGET ERASEUNIT|INITUNIT IS NOT OFFLINE OR NOT CONNECTED OR HAS VOLSER –** The device address specified by `ERASEUNIT=` or `INITUNIT=` is not offline on this system, or has no channel paths, or is marked offline but has a volume serial filled in the UCB. If it is not offline, you can either vary it offline or include the operand `ONLINE=VARYOFF` if you are sure the volume should be erased or initialized. If it is offline with a volume serial number, you may need to vary it online and offline again to clear this illogical condition.

8 **TARGET ERASEUNIT|INITUNIT IS ONLINE AND ACTIVE –** The device address specified by `ERASEUNIT=` or `INITUNIT=` is not offline on this system and it is currently allocated to one or more address spaces, so it cannot be varied offline and erased or initialized.

8 **TARGET ERASEUNIT|INITUNIT FAILED TO COME OFFLINE –** The device address specified by `ERASEUNIT=` or `INITUNIT=` is not offline on this system; an internal `VARY OFFLINE` did not succeed.

8 **TARGET ERASEUNIT CAME ONLINE DURING ERASE –** The device address specified by `ERASEUNIT=` was varied online during the erase process; the process is terminated.

9 **TARGET VOLSER CHANGED –** The volume serial on the target device was changed unexpectedly. One possible cause is an ICKDSF offline INIT of the target device during the swap.

A **SWAP TARGET UNIT ALREADY IN USE –** The specified target device is already in use as a swap target for another source volume. Another FDRPAS SWAP task has this device enqueued and is trying to initiate a swap to it.

B **SOURCE VOLUME DOES NOT CONTAIN VOL1 –** The source volume is online but does not contain a proper z/OS volume label. Use FDRDSF PRINT or another tool to print the label track of the volume and contact INNOVATION Technical Support for assistance.

C **CPU*xxxxx* PAS I/O INTERCEPT FAILED CODE=*c* –** An FDRPAS I/O intercept on the source volume had an internal failure and has de-activated itself. The code "*c*" indicates the cause:

0 – Unknown CCWs (see REASON=E) or illogical condition.

1 – An IOSB had a 64-bit real address (not issued in V5.4/30 or above).

2 – A CCW chain included a TIC to a TIC, an illegal sequence.
4 – Too many concurrent update I/Os on one device. Re-submit this swap when less update activity.
6 – The DDT of the source volume was unexpectedly changed.
8 – An ABEND occurred in the intercept.
9 – Invalid seek address in sense data.
A – GETMAIN failure for ESQA for the FDRPAS track table.
B – Reserved.
C – Cancel by dialog.
D – The ALIAS is disconnected.
E – With `ALLOWPAV=YES` specified, FDRPAS intercepted a write I/O issued to a specific PAV alias. Write I/O's should be done to the base PAV device, not the alias. A PAV command directed to the alias may have caused this error. Re-executing FDRPAS may allow FDRPAS to process this device.
F – The lock cannot be acquired after trying for 3 seconds.
G – The source volume was swapped by HyperSwap.

D **JES3-MANAGED VOLUME FAILED VERIFICATION –** This system is executing under JES3 and the DASD to be swapped is flagged as JES3-managed (DASD referenced by a `DEVICE` statement in the JES3 initialization parameters are JES3-managed). A JES3 routine was called to notify JES3 about the swap, but the routine returned an error. Verify that the target device is offline to JES3.

E ***systemid*** **PAS I/O INTERCEPT FOUND UNKNOWN COMMAND(S) –** The FDRPAS I/O intercepts on the source volume on the indicated system encountered an I/O sequence that they could not interpret. Since the effects are unknown, FDRPAS could not determine if tracks were updated. FDRPAS prints a summary of the unknown commands with the job name of the issuer and the time; please attempt to determine what that job was doing and contact INNOVATION Technical Support so that we can determine if FDRPAS can be enhanced to handle the unknown commands.

NOTE: If the job was using facilities that are documented as restricted during a swap (see Chapter 320 "FDRPAS Special Considerations"), you may be able to resubmit the swap when those facilities are not in use.

F ***systemid*** **FAILED THE SWAP REASON=*x* –** FDRPAS on the indicated system failed the SWAP. "*x*" is an additional reason code, which usually matches the reason code in the **FDR234** message from the system where the failure occurred. "VM" appears at the end of the message if the failure occurred on a VM system. Save the listing from the MONITOR or SWAP task on the indicated system to determine the cause. Contact INNOVATION Technical Support for assistance if necessary.

G **INSUFFICENT STORAGE IN REGION –** Increase the `REGION=` parameter specified in the FDRPAS JCL. `REGION=0M` is recommended. It may also occur if too many FDRPAS tasks are executing in this address space; in that case, reduce the number of concurrent tasks.

H **COULD NOT RAISE IOS LEVEL OR I/O HUNG ON SOURCE –** FDRPAS attempted to raise the IOSLEVEL of the source volume, to inhibit I/Os for a short time during a phase of the swap, but either the IOSLEVEL request failed or there is an active I/O on the device that has not completed within 2 minutes. During Phase 5, it can also occur if one system held a long-term RESERVE or had a long-running I/O on the source volume; you may need to retry the swap during a time when such long I/Os are not likely to occur.

H **COULD NOT RAISE IOS LEVEL VOLUME RESERVED –** FDRPAS attempts to raise the IOSLEVEL of the source volume, to inhibit I/Os for a short time during a phase of the swap, but some other task held a long term RESERVE on the device. Either a task holding the RESERVE is hung; or a process holding a long-term RESERVE, such as a restore of the device, is in progress.

I **I/O INTERCEPTS FAILED TO ACTIVATE –** The dynamic installation of the FDRPAS I/O intercepts on the source volume was unsuccessful.

K **MONITOR SWAP FOUND NO MATCHING OFFLINE UNITS –** None of the device addresses specified on the MOUNT statement following a MONITOR statement were offline DASD devices. This can also occur in FDRERASE if an ERASEUNIT= specifies a 3-digit device address; unit addresses must always be four digits. Note that a MONITOR for a SWAPBUILDIX function requires one offline unit as a communication device.

L **OUR CPUID=*ssssssssss* NOT IN ESTABLISHED PATHS - USING #SYSTEMS= –** The source volume is on a control unit that reports the systems with access to the DASD, but the ID of the system that the FDRPAS SWAP task is executing on is not among those systems. This may indicate a failure in that hardware facility. If you specified the `#SYSTEMS=` operand, FDRPAS prints this message but defaults to that value. If `#SYSTEMS=` is omitted, the step fails. If necessary, contact INNOVATION Technical Support for assistance.

M **CPU (SERIAL#=*serial*) FAILED TO RESPOND IN PHASE *x* –** The system with the indicated CPU serial number did not respond during the indicated phase of FDRPAS operation. This usually results in message FDRW68; see that message for possible causes.
This message is also produced if `DYNMON=YES` is specified on a MONITOR statement for `SWAP TYPE=TRANSIT` since FDRMOVE does not support the use of `DYNMON=YES`.

M **CPU (SERIAL#=*serial*) SOURCE CAME ONLINE DURING SWAP –** When the SWAP operation started, the source volume was not online on the monitor system with the indicated CPU serial number, so that system did not join in the SWAP. Then, while the SWAP operation was in progress, the volume came online to that system. Since that system was not monitoring updates to the volume, the SWAP cannot be completed. A retry should be successful.

N **MORE THAN 128 ACTIVE SYSTEMS OR INTERNAL ERROR –** Either more than 128 systems have access to the source volume or an error occurred determining the number of systems.

N **MONITOR EXCEEDED MAXIMUM NUMBER OF UCBS FOR 1 JOB –** The MOUNT statement following a MONITOR statement specified more than 8190 offline DASD device addresses.

O **CONSISTENCY GROUP MISMATCH SOURCE=*grpname* TARGET=*grpname* –** FDRPAS detected that the source volume was in an EMC consistency group and either the target volume was not in a consistency group, or was in a different group. "*grpname*" may be N/A if FDRPAS was unable to determine the name or if the target is not in a group.

P **ACTIVE PAGE/SWAP DATA SET ON VOLUME –** The volume to be swapped contained an active local page or swap data set.

Q ***function*** **OPERATION CANCELLED BY OPERATOR –** The console operator or a TSO user replied "**NO**" to message **FDRW01, FDRW07, or FDRW68,** or a TSO user requested that the swap be terminated via the FDRPAS ISPF panels. "*function*" is SWAP, ERASE, or INITV.
To avoid issuing these messages as WTORs to the operator, so they can only be replied to from TSO, specify `WTOR=NO` on the main control statement.
To avoid issuing message **FDRW01** or **FDRW07** at all, and proceed with the swap without operator confirmation, do not specify `CONFMESS=YES` on the main control statement.

Q **SWAP OPERATION CANCELLED NONRESPONDING=FAIL –** One or more MONITOR systems failed to respond to a request to start a SWAP. Because `NONRESPONDING=FAIL` was specified on the SWAP or SWAPDUMP command, FDRPAS does not issue message **FDRW68**, but proceeds as though the reply was "NO".

R **SOURCE DOES NOT SUPPORT ACTIVE CPU SERIAL# - SPECIFY #SYSTEMS= –** The source volume is in a DASD subsystem that does not support identifying the serial numbers of the systems that have access to the device (such as a 3990-3, IBM RVA, or Oracle StorageTek SVA). You must specify #SYSTEMS=*nnn* on the SWAP statement to identify the number of systems with access to the DASD.

S **SWAPUNIT DOES NOT SPECIFY 4 CHARACTERS –** The `MOUNT` statement following a `SWAP` or `SWAPDUMP` statement must specify a 4-digit target device address.

T **VOL= ON MOUNT MISSING OR NOT FULL VOLSER –** The `MOUNT` statement following a `SWAP` or `SWAPDUMP` statement must specify one source volume serial, with no asterisk. It may also be that you specified a blank preceding `SWAPUNIT=` in the control statement. The MOUNT statement following a `MONITOR TYPE=VARYONLINE` or `TYPE=RESETVOL` statement must specify `NVOL=`.

U **COMMAND SPECIFIED THAT IS NOT A MOUNT –** Only a `MOUNT` or `EXCLUDE` statement is supported following a `SWAP`, `SWAPDUMP`, or `MONITOR` statement.

V **SWAP VOLUME IS IN A HYPERSWAP ENVIRONMENT –** The specified source volume is currently involved in an IBM GDPS HyperSwap relationship. To swap such volumes requires a special procedure, including `CONFIRMSWAP=YES`. Remove the volume from HyperSwap or see Section 320.6 "FDRPAS and IBM GDPS/PPRC HyperSwap".

W **SWAP SOURCE VOLUME CURRENTLY BEING SWAPPED –** The specified source volume is already being processed by FDRPAS.

X **SWAPUNIT ALREADY IN SWAP STATUS OR PRIOR FAILURE –** The selected target device is already involved in a swap or a prior swap failed in a way that FDRPAS did not completely cleanup the swap. If you are sure that target device is not currently involved in another swap, re-initialize the target DASD with an offline ICKDSF INIT and re-execute the swap.

Y **SWAPUNIT MISMATCHED NUMBER OF CYLINDERS –** The source volume and target device do not have the same number of data cylinders. To accept the swap to a target device with more cylinders, specify `LARGERSIZE=OK` on the `SWAP` statement.

Z **SWAPUNIT DOES NOT SUPPORT LOCATE –** The specified target device does not support the LOCATE RECORD CCW so it is not supported by FDRPAS. This usually means that the device is in a non-caching IBM 3880 subsystem or equivalent.

Z **SWAPUNIT IS RESERVED OR NOT CONFIGURED AS SHARED –** The specified target device has a RESERVE outstanding against it, or it is not marked SHARED in the I/O configuration. If the latter, update the configuration to mark the target devices shared and retry the swap. A RESERVE should not occur against the offline target device; contact INNOVATION Technical Support for assistance diagnosing this error.

Z **SWAPUNIT DOES NOT HAVE A VALID DCE –** FDRPAS cannot locate the DCE (Device Class Extension of the UCB) for the target device. This should not occur; contact INNOVATION Technical Support for assistance.

Action: The swap is terminated. Depending on the error, you may be able to correct the error and re-execute the swap. If in doubt, contact INNOVATION Technical Support for assistance.

FDR235 FDRPAS ON CPU SERIAL# *sssssssss* IS MONITORING THE FOLLOWING *nnnnnn* UNITS:
list of device numbers

Reason: An FDRPAS MONITOR task is monitoring the potential offline target devices listed. It is executing on a system with serial number "*sssssssss*". If some target devices you specified are not listed, it means that those devices were online on this system (MONITOR tasks bypass online DASD).

FDRERASE WILL ERASE THE FOLLOWING *nnnnnn* UNITS:
list of device numbers

Reason: In this form of the **FDR235** message, FDRERASE was requested to erase the indicated DASD devices. If some specified devices were not listed, it probably means that they were not eligible for erase, probably because they are online.

FDRINITV WILL INITV THE FOLLOWING *nnnnnn* UNITS:
list of device numbers

Reason: In this form of the **FDR235** message, FDRINITV was requested to rebuild the VTOCs on the specified offline devices. If some specified devices were not listed, it probably means that they were not eligible for INITV, probably because they are online.

FDR236 *systemid* ACTIVATED I/O INTERCEPTS ON UNIT=*uuuu* [*text*]

Reason: The system indicated has joined in the swap of an online volume on device number "*uuuu*" by installing the FDRPAS I/O intercepts on the device. This message is printed by the MONITOR task on that system. It is also printed for all systems by the SWAP task for the volume. The "*text*" field may contain "**FAILED MISMATCH SOURCE AND TARGET MIDAW FLAGS**". On a System z processor (or beyond), both the source and target devices must support Modified IDAWs (MIDAWs) or both must not. All IBM DASD subsystems support MIDAWs but some non-IBM subsystems may not. In order to swap, you must shut down MIDAW support with the console command: `SETIOS MIDAW=NO`.

systemid* DE-ACTIVATED I/O INTERCEPTS ON UNIT=*uuuu nnnnnnnn* TRACKS UPDATED *internal-info

Reason: This form of the **FDR236** message shows that the system has de-activated the I/O intercepts on device number "*uuuu*". While the intercepts were active, a total of "*nnnnnnnn*" tracks were updated on the volume by this system (this total counts tracks multiple times if they were updated during more than one pass). This message is printed only by the MONITOR task on that system. "internal-info" contains information for use by INNOVATION Technical Support.

FDR237 *systemid* FAILED TO DE-ACTIVATE I/O INTERCEPTS ON UNIT=*uuuu*

Reason: The system indicated attempted to de-activate the I/O intercepts on device number "*uuuu*" but the attempt failed for some reason. Other messages indicate if the swap was successfully completed or not.

Action: The intercepts remain active. In some cases, this causes no harm, but they remain active until the next IPL. However, if the deactivation failed because some other software was trying to modify the DDT of the DASD to install its own intercept, failures may occur. In either case, contact INNOVATION Technical Support for assistance.

FDR238 FDRPAS *startend* MONITOR JOIN TASK FOR UNIT=*uuuu* STC=*procname.taskname* [WITH ERRORS]

Reason: If an FDRPAS MONITOR task is monitoring more than one potential target device, it must start a separate FDRPAS MONITOR task when it detects a swap is beginning on one of those devices. This message documents that such a "join" task has started or ended for device number "*uuuu*". "*startend*" is **STARTED** or **ENDED**. Note that a "**JOIN**" task may not print either message if it was started for a potential swap but determined that a swap was not actually occurring on the target device. If the MONITOR task was started as an external started task, "**STC=**" appears, documenting the name of the FDRPAS PROC (usually PASPROC) and the name of the started task; if the MONITOR task was started as an internal subtask, this does not appear. If a MONITOR task ended with an error condition, the additional text "**WITH ERRORS**" appears.

FDRERASE STARTED ERASE TASK FOR UNIT=*uuuu* CU=*mfrseral*/*ssid*-*diskid*

Reason: In this form of the **FDR238** message, FDRERASE started an internal erase subtask for the indicated DASD device. "*mfr*" is the DASD subsystem manufacturer (for example, IBM), "*seral*" is the subsystem serial number, "*ssid*" is the subsystem ID (SSID), and "*diskid*" is the internal DASD identification, if it can be determined, or the device address, if not.

FDRINITV STARTED INITV TASK FOR UNIT=*uuuu* CU=*mfrseral*/*ssid*-*diskid*

Reason: In this form of the **FDR238** message, FDRINITV started an internal VTOC initialization subtask for the indicated DASD device. "*mfr*" is the DASD subsystem manufacturer (for example, IBM), "*seral*" is the subsystem serial number, "*ssid*" is the subsystem ID (SSID), and "*diskid*" is the internal DASD identification, if it can be determined, or the device address, if not.

FDR239 *nnnnnnn* TRACKS UPDATED BY *systemid*
FDR239 *nnnnnnn* TOTAL UNIQUE TRACKS UPDATED IN PASS *ppppp*
FDR239 - RE-COPYING UPDATED TRACKS

Reason: Once FDRPAS has completed the initial copy phase during a swap of a volume, it re-copies any tracks that were updated on that volume during that phase. If additional tracks are updated during that re-copy pass, those tracks may be copied again during an additional pass.

The first form of the **FDR239** message shows the number of updated tracks that were reported by the MONITOR or SWAP task on the indicated LPAR during one pass. One message is printed per LPAR participating in the swap if the number reported by that LPAR is non-zero.

The second form of the **FDR239** message shows the total tracks that were updated and must be re-copied during the pass; this may be less than the total of the tracks reported by each LPAR if the same tracks were updated by more than one LPAR.

Action: These messages repeat, and the pass number increments, until the total number of tracks to be re-copied falls below a threshold, at that point FDRPAS completes the swap.

- CONFIRMSWAP IN EFFECT

Reason: If this additional text is displayed, `CONFIRMSWAP=YES` is in effect for the volume.

Action: When the number of tracks to be re-copied falls below the threshold, these messages repeat, and the pass number increments, until you indicate that FDRPAS is to complete the swap via the FDRPAS ISPF interface (see Section 310.31 "FDRPAS ISPF Interface") or a `TYPE=CONFIRMSWAP` batch job (see Section 310.7 "MONITOR CONFIRM Statement").

- CFW/CC DELAY IN EFFECT

Reason: If this additional text is displayed, FDRPAS detected that some job is using Cache Fast Write (CFW) or Concurrent Copy (CC) on the volume.

Action: If the volume is swapped while Cache Fast Write (CFW) or Concurrent Copy (CC) is in use, the CFW/CC job may fail, so FDRPAS delays the swap up to 2 minutes to see if the CFW/CC user finishes. Note that if no Cache Fast Write (CFW) or Concurrent Copy (CC) I/O is issued for a few minutes, FDRPAS may think it is done, or may never detect the CFW/CC usage at all, so it is still possible for the job to fail after the swap is complete.

- WAIT FOR HYPERSWAP DISABLE

- WAIT FOR AUTOSWAP DISABLE

Reason: If this additional text is displayed, `CONFIRMSWAP=YES` was specified so FDRPAS allowed a swap of a volume enabled for IBM HyperSwap or EMC AutoSwap to proceed. However, it waits before the final swap until HyperSwap or AutoSwap is disabled on the DASD.

Action: Disable HyperSwap or AutoSwap. See Section 320.5 "FDR Extended MCS Software Console (FDREMCS)" through Section 320.8 "FDRPAS and EMC AutoSwap" for further information.

- WAIT FOR PAV DISABLE

Reason: If this additional text is displayed, `ALLOWPAV=YES` was specified so FDRPAS did not disable Parallel Access Volume (PAV) aliases until just before the actual swap of the volume. The disabling of the aliases caused a slight delay. This condition should be transient.

Action: If this message appears more than once for a given volume, contact INNOVATION Technical Support for assistance.

- *systemid* NOT RESPONDING

Reason: If this additional text is displayed, the indicated system did not respond at the end of a phase 3 copy pass when the number of tracks to be re-copied was below the threshold. FDRPAS recopies any updated tracks reported by the responding systems and waits some extra time, it then tries again to communicate with all of the monitors. If this condition persists for a number of tries, FDRPAS fails the SWAP.

Action: Check whether the indicated system has adequate resources (for example LPAR weight or capping) to participate in the SWAP.

SWAP OF VOL=*vvvvvv* DELAYED DUE TO DYNAMIC CONFIGURATION ON *system*

Reason: This form of the **FDR239** message is issued when a dynamic I/O re-configuration (the console ACTIVATE command) is in progress. The final swap of the volume cannot be done until the ACTIVATE function completes, so FDRPAS delays until the ACTIVATE is done.

Action: If this condition persists for more than a few minutes, the ACTIVATE command completion may be delayed for some reason. You should investigate the delay and resolve the cause so that FDRPAS swaps can continue.

FDR240 SWAP FOR VOL=*vvvvvv* WAS ABNORMALLY TERMINATED BY *systemid*

SWAP FOR VOL=*vvvvvv* WAS CANCELLED BY OPERATOR ON *systemid*

Reason: The swap of volume "*vvvvvv*" was terminated by an action, either an abnormal termination of FDRPAS or a request to terminate, on the indicated system. "CANCELLED" appears only if the swap was terminated via the FDRPAS ISPF interface. Check the listing of the MONITOR task from that system for the reason.

FDR241 FDRPAS SUCCESSFULLY COMPLETED SWAP OF VOL=*vvvvvv* TO UNIT=*uuuu* ON *systemid*

Reason: FDRPAS has completed the swap of volume "*vvvvvv*" to unit "*uuuu*" on the indicated system. This message prints in the MONITOR task output on each system, and the message from each system also prints in the SWAP task output.

FDRERASE SUCCESSFULLY COMPLETED *function* OF VOL=*vvvvvv* ON UNIT=*uuuu*

Reason: In this form of the **FDR241** message, FDRERASE has finished erasing (ERASE) or emptying (EMPTY) volume "*vvvvvv*" (which may be blank or asterisks if the volume serial number is unknown) on unit "*uuuu*".

FDRINITV SUCCESSFULLY COMPLETED INITV OF VOL=*vvvvvv* ON UNIT=*uuuu*

Reason: In this form of the **FDR241** message, FDRINITV has finished initializing volume "*vvvvvv*" on unit "*uuuu*".

FDR242 ADDRESS SPACE CREATE TO JOIN FAILED FOR UNIT=*uuuu* [COMP=*cccc*]

Reason: If an FDRPAS MONITOR task is monitoring more than one potential target device, it must start a separate FDRPAS MONITOR task when it detects a swap is beginning on one of those devices. However, the address space creation for a "join" task for unit "*uuuu*" failed with return code "*cccc*". If "**COMP=***cccc*"is not displayed, one possible reason is that you have not placed the PASPROC JCL member required in an appropriate system procedure library, or you have renamed it and not updated the name in the FDR Global Options Table (see Section 380.16 "FDRPAS and FDRERASE Options (Panel A.I.4.14)"). A return code of 52 indicates that the system was unable to create an address space for the "join" task, because of system resource limitations or the system-wide limit on address spaces is exceeded.

Action: Check SYSLOG for messages to determine the cause. If possible, correct the error and re-submit the FDRPAS request.

FDR243 SWAP OF UCB ADDRESSES FAILED VOL=*vvvvvv* COMP=*ccccc*

Reason: FDRPAS has invoked a system service to swap the Unit Control Blocks (UCBs) of the indicated source volume and its target device, but that service failed with return code "*ccccc*". If this failure occurs on a system running an FDRPAS MONITOR task, this message is printed only in the output of that MONITOR task, but the SWAP task prints a related message indicating the failure on this system.

Action: Check the output of the SWAP and MONITOR tasks on all systems. If the swap failed on all systems, then the swap was terminated and no harm was done. If the swap completed on one or more systems, then you should stop all usage of the volume on the failing systems, since updates are now being directed to the wrong device, and contact INNOVATION Technical Support immediately for assistance.

FDR244 *systemid* FAILED TO ACKNOWLEDGE SWAP COMPLETION VOL=*vvvvvv*

Reason: FDRPAS attempted to complete the swap operation, but the system indicated did not acknowledge that the swap was complete. This may indicate that system has become nonoperational during the swap, or that the FDRPAS MONITOR task on that system has failed, but the most likely cause is that the indicated system is a low-priority LPAR or the MONITOR task is running at low priority, so that it was not able to acknowledge the swap within two minutes. The swap may have completed on some systems. This message is printed in the output of the SWAP task.

Action: Check the messages from the FDRPAS MONITOR task on the indicated system(s). If the messages indicate that the swap was completed on all systems, then the swap was successful and no action must be taken. In the unlikely situation that the swap did not complete or the MONITOR task was terminated prematurely on one or more systems, then you should stop all usage of the volume on the failing systems, since updates are now being directed to the wrong device, and contact INNOVATION Technical Support immediately for assistance.

FDR245 FDRPAS CONFIRMED THE SWAP OF VOL=*vvvvvv*

Reason: A `MONITOR TYPE=CONFIRMSWAP` or `TYPE=CONFIRMSPLIT` was submitted for one or more volumes. When all the specified volumes are ready for completion, FDRPAS completes the SWAP or SWAPDUMP and issues the FDR245 message for each volume.

FDR246 SWAP NOT ACTIVE FOR CONFIRM OF VOL=*vvvvvv*

Reason: A `MONITOR TYPE=CONFIRMSWAP` or `TYPE=CONFIRMSPLIT` was submitted for one or more volumes, but the volume indicated was not currently involved in a SWAP or SWAPDUMP operation with `CONFIRMSWAP=YES` or `CONFIRMSPLIT=YES` specified.

Action: Correct the list of volumes specified on the `MOUNT` statements to include only those actively being swapped and re-submit the job.

FDR247 UNABLE TO CATALOG SWAP HISTORY RECORD COMP=X'*FFFF00001111*'

Reason: After a swap, FDRPAS attempted to catalog a history record to document the swap, but the catalog request failed. COMP contains the catalog return codes from registers R15 (*FFFF*), R0 (*0000*), and R1 (*1111*). The swap was successful but it was not recorded for historical and reporting purposes.

Action: The history record is cataloged into the system catalog with an alias equal to the value of PASINDEX (see "PASINDEX" in Section 380.16). The catalog may be full. If so, run the example "Display and Purge History for All Volumes Example" in Section 310.25 to purge some old history records. If you can not find any problem with the catalog, contact INNOVATION Technical Support for assistance.

UNABLE TO CATALOG SWAP HISTORY RECORD - NO ALIAS FOR PASINDEX

Reason: This form of the **FDR247** message indicates that after a swap, FDRPAS attempted to catalog a history record to document the swap, but there was no alias defined in the master catalog of this system matching PASINDEX (see "PASINDEX" in Section 380.16). The swap was successful but it was not recorded for historical and reporting purposes.

Action: If you wish history records to be recorded on this system, define a user catalog (if necessary) and associated an alias matching PASINDEX with that catalog.

UNABLE TO CATALOG SWAP HISTORY RECORD - CATALOG ON VOLUME BEING SWAPPED

Reason: This form of the **FDR247** message indicates that after a swap, FDRPAS attempted to catalog a history record to document the swap, but the catalog pointed to by the alias matching PASINDEX (see "PASINDEX" in Section 380.16) was on the volume being swapped, so FDRPAS is unable to record this swap. The swap was successful but it was not recorded for historical and reporting purposes.

UNABLE TO CATALOG SWAP HISTORY RECORD - USER BYPASSED HISTORY

Reason: This form of the **FDR247** message indicates that the user requested that FDRPAS not cut a history record. The swap was successful but it was not recorded for historical and reporting purposes.

FDR248 NO OFFLINE UNITS LEFT TO MONITOR

Reason: An FDRPAS MONITOR task discovered that all of the offline units it was monitoring are now online or are otherwise ineligible for monitoring. This may occur if FDRPAS SWAP tasks have swapped volumes to all of those offline units.

Action: The MONITOR task terminates normally.

FDR249 WARNING: *type* VOLUME SWAPPED TO NON-type VOL=*volser*

Reason: You requested that volume "*volser*" be swapped, but FDRPAS detected that the source volume is currently being mirrored by a hardware facility such as PPRC, but the target volume does not have the same sort of mirroring. Currently FDRPAS checks for PPRC mirroring and PPRC-compatible facilities.

Action: The swap continues, but this message warns you that after the swap completes, the volume is no longer protected by a remote mirror. If your disaster recovery plan or other needs require that this volume be remotely mirrored, you need to re-establish remote mirroring after the swap is complete.

FDR250 CYL=*ccccccc xxxx xxxx … xxxx*

Reason: Internal message showing each cylinder and track updated on this system during a Phase 3 pass; it appears only if `PRINT=ALL` is specified on the `SWAP` or `MONITOR` statement. "*ccccccc*" is a cylinder number in decimal. "*xxxx*" displays a set of 16 bits in hex; the first 15 bits show which tracks were updated. The first "*xxxx*" is for cylinder "*ccccccc*", the next for cylinder "*ccccccc*"+1, and so on, up to "*ccccccc*"+19, thus showing 20 cylinders in each **FDR250** message.

FDR251 FDR UNBOUND PAV ALIASES FROM UNIT=*uuuu* [*text*]

Reason: In this form of the **FDR251** message, unit "*uuuu*", which is the source or target device of a swap, had Parallel Access Volume (PAV) aliases bound to it. FDRPAS has unbound the aliases and disabled PAV on the device during the swap.

FDR RE-ENABLED PAV FOR UNIT=*uuuu* [*text*]

Reason: In this form of the **FDR251** message, unit "*uuuu*", which is the source or target device of a swap, has Parallel Access Volume (PAV) re-enabled. If it had static PAV aliases assigned, they have been rebound to the base device. This is done only if both the source and target devices in a swap were enabled for PAV before the swap.

In either form of the **FDR251** message, it may be followed by the text "**FAILED COMP=**" to indicate that the IBM service that was called for PAV has failed with the indicated return codes.

FDR252 FDR SWAPPED IPLABLE UNIT=*uuu1* TO *uuu2* YOU MUST UPDATE YOUR HARDWARE MANAGEMENT CONSOLE

Reason: Unit "*uuu1*", which is the source volume of a swap, is an IPLable volume, meaning that it had IPL text or an IODF data set on it. This may mean it was the system residence (SYSRES) volume for a z/OS system, or it may mean it contained an IPLable utility such as ICKDSF, DFSMSdss, SADMP, INNOVATION DATA PROCESSING's Stand Alone Restore (SAR), or NewEra Software's SAE. It has been swapped to unit "*uuu2*". This message is issued as a nonscrollable console message as well as printed in the FDRPAS output.

Action: If it was a SYSRES, you must update the IPL or IODF address in the system startup parameters on the Hardware Management Console (HMC) or hardware console for that system. If it contained an IPLable utility, you should update appropriate documentation with the new IPL address.

FDR253 REASON=*reason*

Reason: An error occurred when FDRPAS was trying to activate or de-activate its I/O intercepts on a source volume. This message is issued only as a WTO to the operator but it also appears in the job log of the FDRPAS job. Reason codes include:

- **1** **PASV ID ERROR –** Internal installation error. Contact INNOVATION Technical Support for assistance.
- **2** **PAS DDT FIND ERROR –** Internal installation error. Contact INNOVATION Technical Support for assistance.
- **3** **GETMAIN ERROR –** An error occurred doing a GETMAIN for storage in ECSA for the intercepts.
- **4** **NO DCE ERROR –** The source volume UCB does not point to a DCE (Device Class Extension). This should not be possible for an online DASD.
- **5** **DCE LENGTH ERROR –** The DCE (Device Class Extension) pointed to by the UCB of the source volume is not at least 48 bytes in length. This should not be possible.
- **6** **DCEALCYL INVALID ERROR –** Number of alternate cylinders in the DCE (Device Class Extension) of the source volume is not valid. This may indicate that the source volume is on an old control unit not supported by FDRPAS.
- **7** **XTID VALIDATION ERROR –** Internal validation or coordination error. Contact INNOVATION Technical Support for assistance.
- **8** **NAME/TOKEN NOT FOUND –** Internal validation or coordination error. Contact INNOVATION Technical Support for assistance.
- **9** **NAME/TOKEN CREATE ERROR –** Internal validation or coordination error. Contact INNOVATION Technical Support for assistance.
- **A** **NAME/TOKEN DELETE ERROR –** Internal validation or coordination error. Contact INNOVATION Technical Support for assistance.
- **B** **TOKEN ERROR=0 –** Internal validation or coordination error. Contact INNOVATION Technical Support for assistance.
- **C** **NAME/TOKEN RETRVE ERROR –** Internal validation or coordination error. Contact INNOVATION Technical Support for assistance.
- **D** **LOAD ERROR –** An error occurred loading the intercept module FDRPASXT.
- **E** **BLDL ERROR –** An error occurred during a BLDL on the intercept module FDRPASXT.
- **F** **UCBDDT CHANGED ERROR –** The pointer to the DDT (Device Descriptor Table) in the UCB of the source volume changed unexpectedly.
- **F** **UCB ERROR, UNIT SWAPPED BY OTHER –** The source volume was swapped by HyperSwap.

G **UCB BUSY OR IOSLEVEL NOT RAISED –** Either the IOSLEVEL function failed to raise the IOS level of the source volume, or the source volume was continuously busy for 2 seconds.

H **ABNORMAL EXIT ADDR NOT RESET –** Internal installation error. Contact INNOVATION Technical Support for assistance.

I **DDT NOT WITHIN IGGDDT* –** Internal installation error. Contact INNOVATION Technical Support for assistance.

J **UNABLE TO FIND MATCHING MIHB –** For either the source volume or target device, FDRPAS was unable to find an entry in the MIHB (MIH table) with a key value matching the UCBMIHKY value in the associated UCB. Since this would cause a failure when the volume is swapped, the swap is not attempted. You may be able to use the console command "`SET IOS=`*xx*" to rebuild the MIHB and retry the swap.

K **MIDAW MISMATCH –** On a System z processor (or beyond), both the source and target devices must support MIDAWs (Modified IDAWs) or both must not. All IBM DASD subsystems support MIDAWs but some non-IBM subsystems may not. In order to swap, you must shut down MIDAW support with the console command: `SETIOS MIDAW=NO`.

Action: The swap is terminated. Contact INNOVATION Technical Support for assistance.

FDR254 INDEXED VTOC BUILD COMPLETE

Reason: A SWAPBUILDIX function completed successfully. This form of the **FDR254** message is issued by the SWAPBUILDIX task.

INDEXED VTOC REFRESHED

Reason: A SWAPBUILDIX function completed successfully. This form of the **FDR254** message is issued by the MONITOR tasks participating in the function to indicate that the indexed VTOC information has been updated on this system.

INDEXED VTOC BUILD FAILED - CHECK IXSYSPRT

Reason: A SWAPBUILDIX function failed.

Action: An IXSYSPRT DD statement was dynamically allocated to SYSOUT; check it for messages from ICKDSF.

ICKDSF MSG=ICK516I *uuuu* I/O ERROR DETECTED DURING VTOC CONVERSION: ERROR CODE= 5

Reason: A FDRPAS function failed because there was insufficient space for the indexed VTOC.

Action: A 3390-27 requires a minimum of two tracks for the Indexed VTOC. A 3390-54 and a 3390-A EAV require a minimum of four tracks for the Indexed VTOC. Ensure the minimum sizes exist before running the FDRPAS function.

FDR255 *function* TERMINATED TARGET UNIT *uuuu* CONTAINS DATA SETS FIRST DSN=*dsname*

Reason: `CHECKTARGET=YES` was specified, and the offline target DASD designated for a SWAP, SWAPDUMP, ERASE, or INITV was not empty. It contained at least one data set other than the VTOC, VTOCIX, and VVDS. The first data set name encountered in the VTOC is listed. If the volume was formatted by z/VM (with a dummy VTOC), "*dsname*" is "**VM VOLUME**".

Action: Check the target DASD to see if it contains data that must be preserved; you may need to vary the DASD online in order to list its VTOC. If so, choose another target for the swap or erase. If not, remove the `CHECKTARGET=YES` operand (or specify `CHECKTARGET=NO`) to overlay the DASD.

function TERMINATED TARGET UNIT *uuuu* VOL=*vvvvvv* DOES NOT MATCH CHKTVOL

Reason: This form of the **FDR255** message is issued when CHECKTARGETVOL= is specified on a MOUNT, and the offline target DASD designated for a SWAP, SWAPDUMP, or ERASE does not have the specified volume serial. "*vvvvvv*" is the volume serial actually found on the DASD.

Action: Check the target DASD to see if it is actually the DASD you want to erase or use as an FDRPAS target. If so, remove the CHECKTARGETVOL= operand or specify the actual volume serial to overlay the DASD.

FDR256 REPORT FROM SYSPRIN*x*

Reason: MAXTASKS= was specified. Messages from swap subtasks are written to SYSPRIN*x* DD statements, but they are also echoed in SYSPRINT with this header when each swap terminates.

FDR257 DYNAMIC PACE DELAY CHANGED TO *sss.ss* SEC - I/O DELAY *mmmmm* MSEC

Reason: PACING=DYNAMIC was specified. Every 15 seconds FDRPAS may issue this message to indicate that the I/O pacing value was changed to "*sss.ss*" seconds because the average I/O delay, as determined by an internal FDRPAS algorithm, has changed to "*mmmmm*" milliseconds.

FDR258 *e-mail error text*

Reason: The FDREMAIL DD statement was provided to invoke the FDR e-mail facility, but an error occurred preventing an e-mail from being sent. The message is printed on SYSPRINT and is also sent via a Write-To-Log (WTL) macro, so it appears in the job log of the FDRPAS job, and may appear on a system console. The message text defines the error condition. It may contain TCP/IP error codes that can be found in IBM Communication Server manuals.

Action: FDRPAS continues if it has more volumes to process. Correct the error for future jobs.

FDR259 EMAIL MESSAGE(S) SENT

Reason: The FDREMAIL DD statement was provided to invoke the FDR e-mail facility, and one or more e-mail messages were successfully sent to the mail server you specified. Note that this is no guarantee that the messages are delivered if e-mail addresses are invalid or other errors occur after the mail server accepts the message.

FDR260 VARY ONLINE FAILED CODE=*cccc rrrr ssss message*

Reason: FDRPAS called the IBM service IEEVARYD to initialize system control blocks for the target device, and the VARY failed with the codes and/or message shown. "*cccc*" is the return code from IEEVARYD. If it is 0004, then "*rrrr ssss*" are additional return and reason codes that are documented in the IBM macro IEEZB834. If IEEVARYD returned additional message text, it is displayed at the end.

If "*cccc rrrr*" is **0016 0032**, this is a code generated by FDRPAS to indicate that the IEEVARYD function was terminated because it took an excessive amount of time.

Action: Despite the error, the target device is online and usable. Contact INNOVATION Technical Support for assistance in determining if there are any considerations due to the error.

VARY ONLINE FAILED UNIT=*uuuu*

Reason: This form of the **FDR260** message indicates that a `MONITOR TYPE=VARYONLINE` or `TYPE=RESETVOL` was executed but either the requested device was not eligible (already online or does not have a volume label set by FDRPAS) or, if the additional text **"VOLSER ALREADY ONLINE"** appears, the new volume serial requested by `NVOL=` is already online on another device.

Action: Verify that `VARYUNIT=` specifies the address of an offline DASD that was a SWAP source volume or a SWAPDUMP target device. If the volume serial number was already online, modify the `NVOL=` to create a unique volume serial number or run the function on a system where the volume is not online.

FDR261 UNIT=*uuuu* VOL=*vvvvvv* WAS *function*

VARY FAILED COMP=*nn* CODE=*cc* CHECK MVS LOG [PENDING OFFLINE]

Reason: This message can be issued by an FDRPAS MONITOR, by FDRERASE, or by FDRINITV. "*function*" can be "**VARIED ONLINE**", **"VARIED OFFLINE"**, or **"RESET"**.

The second form of this message can also indicate that a VARYON or VARYOFF failed.

FDR262 MODULE ICKDSF NOT FOUND OR INSUFFICIENT AUTHORITY ABEND=S*cde-rc*

Reason: `LARGERSIZE=OK` or `SMALLERSIZE=OK` was coded. FDRPAS needs to call ICKDSF to rebuild the indexed VTOC when a volume is swapped to a larger or smaller DASD, so a test LOAD of ICKDSF was done to be sure that it is available and this job is authorized to use it. The load failed with the system ABEND code (*cde*) and reason code (*rc*) indicated.

Action: The FDRPAS step terminated with an U0502 ABEND. Authorize the FDRPAS job to use ICKDSF or correct other errors. If the target devices are all the size as their source volumes, those operands are not required and can be removed to avoid the error.

FDR263 CSA/SQA SHORTAGE DELAY IN EFFECT

Reason: FDRPAS detected that available below-the-line SQA and CSA storage has fallen below a threshold, possibly due to IOS/EXCP control blocks in SQA created by the I/Os issued by FDRPAS to copy data.

Action: FDRPAS reduces its rate of I/O until the CSA/SQA available storage rises again. If this message occurs frequently, you may need to reduce the number of concurrent FDRPAS operations or increase the size of your below-the-line SQA.

FDR264 FDR {DISABLED|RE-ENABLED} FLASHCOPY UNIT=*uuuu* [FAILED COMP=*nnnn*]

Reason: Unit "*uuuu*", which is the source volume of a swap, was capable of FlashCopy. The DISABLED form of the message is issued to indicate that FDRPAS has disabled FlashCopy at the beginning of the swap, since FDRPAS cannot detect tracks updated by FlashCopy. The RE-ENABLED form is issued at the end of the swap to indicate that FlashCopy has been re-enabled. Either form of the message may be followed by the text "**FAILED COMP=**" to indicate that the IBM service that was called for FlashCopy has failed with the indicated return codes.

Action: Any attempt to initiate a FlashCopy to the FDRPAS source volume during the swap fails. Most products that implement FlashCopy revert to normal read/write I/O if FlashCopy cannot be used; this includes FDRCOPY.

FDR265 STOP COMMAND ACCEPTED WAITING ACTIVE TASK TERMINATION
STOP COMMAND ACCEPTED DUE TO STOPAFTER TIME EXPIRED
STOP COMMAND ACCEPTED DUE TO STOPINACTIVE TIME EXPIRED

Reason: A console `STOP (P)` command was issued for an FDRPAS, FDRMOVE, or FDRERASE job or started task or the time limit specified in either the `STOPINACTIVE=` or `STOPAFTER=` operand has been reached.

Action: A MONITOR task that is STOPPED terminates within a few seconds if it is not participating in any active swaps; otherwise, it terminates when the active swaps have completed; no new swaps are started. If a SWAP task is STOPPED, it terminates when all active swaps have completed; no new swaps are started. FDRERASE stops when all active erases have completed.

FDR266 SWAP BYPASSED DUE TO STOP COMMAND VOL=*vvvvvv*

Reason: A console `STOP (P)` command was issued for an FDRPAS job or started task. This documents that a swap for volume "*vvvvvv*" was bypassed because of the `STOP (P)`.

ERASE BYPASSED DUE TO STOP COMMAND UNIT=*uuuu*

Reason: This form of the **FDR266** message indicates that a console `STOP (P)` command was issued for an FDRERASE job. This documents that an erase for DASD device "*uuuu*" was bypassed because of the `STOP (P)`.

FDR267 CANCEL PROTECTION IN EFFECT - WAITING FOR TASK END - ISSUE 2ND CANCEL FOR IMMEDIATE TERMINATION

Reason: A console `CANCEL (C)` command was issued for an FDRPAS job or started task and cancel protection (`CANCELPROT=YES`) was in effect. Cancel protection is the default for MONITOR tasks.

Action: With cancel protection active, a `CANCEL (C)` acts like a `STOP (P)` command; see message FDR265 above for details. If any swaps are active, it waits for them to complete. If you need to immediately terminate the task, enter a `CANCEL (C)` command again.

FDR268 VOL=*vvvvvv* ERROR - *reason text*

Reason: An FDRPAS license report (`LICENSE TYPE=SWAP`) encountered an error determining the size or the free space on the indicated volume. "*reason text*" details the error.

Action: The volume is not included in the license report.

FDR302 CONTROL STATEMENT ERROR NEAR REL LOCATION *nn* -- REASON *x* -- JOB TERMINATED

Reason: An error was encountered during the processing of a user-supplied control statement. If "**NEAR REL LOCATION *nn***" appears, the keyword or operand causing the error is at or near column "*nn*" on the input statement. The error is defined by the reason code within the message. The failing statement is displayed immediately above.

NOTE: The expression "SELECT statement" in the reason codes below refers to MOUNT and EXCLUDE statements.

The reason "*x*" can be:

1 – A MOUNT statement did not specify any operands. Control statement was blank after the Command name.
2 – Command name on the first control statement was incorrectly specified. It must be `SWAP`, `SWAPDUMP`, or `MONITOR`.
3 – Operand on the first control statement was incorrectly specified.
4 – Operand did not end with a blank or comma.
5 – SYSIN data set was empty.

6 – Expected continuation statement was not found. The previous statement ended with a comma and a blank.

7 – One of the following:

- On the first control statement, invalid or incompatible operands were specified.
- The TYPE= operand was omitted on SWAP, SWAPDUMP, or MONITOR statement.

8 – An operand on a MOUNT statement specified a blank or comma after the equal sign.

9 – On the control statement printed above, one of the options exceeded its maximum length, or was shorter than its minimum length. On a MOUNT statement for MONITOR, more than 255 addresses or address masks were specified.

C – Maximum number of MOUNT and EXCLUDE statements was exceeded. The limit is 250 or the value specified for MAXCARDS=.

F – An operand that requires numeric data (for example, MAXCARDS=) specified non-numeric characters, or an operand that requires hexadecimal data (for example, SWAPUNIT=) specified non-hex characters.

G – An operand did not end with a blank or comma or exceeded 15 digits (for example, MAXCARDS=).

I – Keyword is invalid under the operation indicated.

J – Control statement was completely blank. You can enter comment lines by placing an * (asterisk) in column 1.

K – A required operand was not specified on the preceding statement.

Q – Keyword exceeded maximum value or was negative.

R – TYPE=*xxx* was specified multiple times.

S – An operand on the MOUNT statement was specified multiple times or was mutually exclusive with another operand.

U – The TYPE=*xxx* operand was missing or invalid on the first statement. This operand is required.

Action: Correct error and resubmit job. If you do not understand the error after reviewing the relevant sections of this manual, call INNOVATION Technical Support for assistance.

HINT: *If the control statements look good and you can't see any obvious reason for the error, check the JCL to be sure you are executing the right program for the statements you provided (PGM=FDRPAS).*

FDR302 SERVICE CLASS CHANGED TO *classname* FAILED REASON=*retcode*-*reasoncode*

Reason: FDRPAS encountered an error when it attempted to reset its service class. "*retcode*" is the return code and "*reasoncode*" is the reason code from the IWMRESET macro, as documented in the appropriate IBM manual, such as *MVS Programming: Workload Management Services*. The most likely error is 000C-*xxxx*0C28, indicating that the SRVCLASS=*classname* parameter specified a service class that is not defined.

Action: Specify a service class name that is defined to the active workload manager policy.

FDR303 CARD IMAGE - *control statement image source*

Reason: An input control statement is displayed by this message. "*source*" may be:

blank **–** From SYSIN.
PARM ENTRY – From the JCL PARM= operand.

FDR303 SERVICE CLASS CHANGED TO *srvclass*

Reason: The service class was changed to "*classname*" for the MONITOR or SWAP task.

FDR311 FDR MOVED DSN=*dsname status*
ON VOLSER=*volser* UNIT=*device* STORCLAS=*sc*
MGMTCLAS=*mc* DATACLAS=*dc*
CLUSTER=*cluster* NEWC=*newcluster*

Reason: The non-VSAM data sets or VSAM component "*dsname*" was MOVED. The "*status*" also includes:

ALLOCATED – The data set was allocated by FDRMOVE.
CATALOGED – The data set was cataloged by FDRMOVE.
INSTANT – A hardware facility was used to quickly copy data. This occurs only if you are licensed for FDRINSTANT or FDRMOVE. On an IBM Shark or DS8000 or HDS with the FlashCopy feature, FlashCopy was used. On an EMC Symmetrix, an internal EMC copy function was used. Both the input and output volumes must be in the same subsystem (EMC, HDS, or IBM).

"*volser*" defines the volume serial of the DASD where it was moved, and "*device*" identifies the DASD type and model (for example, 3390-3). If the output data set is SMS-managed, the SMS classes assigned to the data set are displayed (management class and data class may be "(NULL)" if not assigned).

For a VSAM component, the line containing "*cluster*" is displayed to identify the cluster to which the component belongs.

FDR313 *fff* TERMINATED BY OPEN EXIT ON VOL= *vvvvvv*

Reason: The locally-written FDR volume open exit has terminated processing of the volume "*vvvvvv*".

FDR314 FDRMOVE *dsn* SCRATCHED FROM VOL=*vvvvvv*

Reason: FDRMOVE documents that one non-VSAM data set, VSAM component, or VSAM cluster was scratched.

FDR316 FDR DID NOT FIND REQUESTED MOUNT SWAP VOL=*vvvvvv*

Reason: A `MOUNT` statement was specified for the indicated volume, but that volume was not online to the system were FDRPAS was run.

Action: Correct the `MOUNT` statement and resubmit, if necessary.

FDR319 FDR OPERATION ABNORMALLY TERMINATED VOL=*vvvvvv* COMP CODE=S*sss*
U*uuuu* [*WILL RETRY WITH MAXTASKS REDUCED*]

Reason: An internal FDR subtask failed on volume "*vvvvvv*".

If "*sss*" is non-zero (in hex), the task failed with a System Ssss ABEND. Consult IBM documentation for the meaning of the ABEND. S*x*13/S*x*14 ABENDs are OPEN/CLOSE errors and are accompanied by an explanatory message in the joblog of the FDR job.

If "*uuuu*" is non-zero (in decimal), the task failed with a User U*uuuu* ABEND. The ABEND is probably generated by FDRPAS.

Action: If the problem cannot be determined from the ABEND code, contact INNOVATION Technical Support for assistance. If the ABEND produced a storage dump, have it available.

If "***WILL RETRY WITH MAXTASKS REDUCED***" is appended to the message, the S087--10 ABEND was intercepted and the task was rerun with the `MAXTASKS=` value reduced. The job successfully moves all the data sets when this form of the message is given.

FDR320 UNABLE TO *function* REASON=*x*--DSNAME=*dsname*

Reason: FDR was unable to RESTORE, COPY, or MOVE the data set specified in the message. The reason code "*x*" may be:

1 – The FDR restore subtask restoring this data set from a backup failed; see message FDR319, other FDR messages may also appear.

3 – I/O error reading a Format 2 or Format 3 DSCB from the output DASD.

6 – Data set enqueue failed for this *dsname*. Specify `DSNENQ=NONE` to override this check.

B – Data set is unmovable.

C – Data set not found on the backup file. The user may have misidentified the backup (the backup does not contain the data set), or it may be due to:

- The same dsname was specified on more than one control statement or the dsname is a component of previously requested VSAM cluster.
- For VSAM clusters, can occur if there was more than one VVR for a component of the cluster in the VVDS of the original DASD when dumped (this is an error).

D – Data set has an F3 DSCB, but it could not be found on the backup.

E – Data set was being restored to a new name and the FDR security checking option (ALLCALL) is enabled but:

- The user does not have READ authority to the original data set.
- The original data set is not protected by your security system (this is considered an error since the data set may have been backed up on another system where it was protected).

G – Unable to locate data set's F3 DSCB on target volume or F2 DSCB found for a non-ISAM data set.

J – Data set was RACF or password protected when it was backed up, but user has preallocated it without protection, or vice versa.

K – Data set has not reached its expiration date, and the operator replied "NO" to the **FDRW03** message.

O – PRESTAGE was specified. Data set restore was bypassed since data set was found.

P – Characteristics of the input data set and the preallocated output data set do not match. Examples of this:

- VSAM to non-VSAM
- PDSE to non-PDSE
- HFS to non-HFS
- Extended Format SMS to non-EF
- And vice-versa in each case. Other combinations also cause errors.

S – A data set that must be SMS-managed (such as Extended Format (EF)) is being restored to a non SMS-managed volume.

T – FDRCOPY COPY/MOVE detected a copy/move of a data set on top of itself (same volume serial number, same data set name).

U – Error updating ISAM Format 2 DSCB, or VSAM component in the VTOC but not in the VVDS.

V – VSAM cluster is being restored to a VSAM cluster that has different options or characteristics. Message FDR152 or FDR159 details the error.

W – Catalog inaccessible. Possible causes are:

- When restoring a catalog, the JCL contained a STEPCAT or JOBCAT DD statement. This is not supported.
- When restoring a catalog, the catalog being restored was defined in the master catalog, but it was not usable (probably did not actually exist on DASD). You must remove the entry for the old catalog, probably by IDCAMS `EXPORT DISCONNECT`.

X – Data set security exit rejected this data set.

Y – NEWINDEX= failure or duplicate data set name generated. It may be that the new data set name does not meet IBM standards, or that the resulting name is greater than 44 characters.

Z – One or more data sets failed to restore using ALLDSN or DSN=*mask*. Check the subtask listing for errors.

Action: Take appropriate action and/or contact INNOVATION Technical Support for assistance.

FDR332 *function* SCRATCH BYPASSED ON DSN=*dsname*

Reason: An attempt to delete the source data set was BYPASSED by the FDRMOVE operation. The scratch was not attempted for this data set because of an earlier error, such as a catalog error, which is described by another message.

FDR334 FDR FAILED TO CATALOG *comp* DSN=*dsname status*

Reason: FDRMOVE allocated and moved the non-VSAM data set "*dsname*", but could not catalog it.

If "*comp*" is printed, it is **COMP=X'***FFFF00001111***'** that contains the catalog return codes from registers R15 (*FFFF*), R0 (*0000*), and R1 (*1111*).

If the register 15 value is **X'FF14'**, then the code was generated by FDR, and indicates either that the data set was currently cataloged on zero volumes or more than 255 volumes, or that the volume sequence number of the current piece of the data set was higher than 255; the R0 value is the number of volumes currently in the catalog entry (in hex). FDR supports cataloging of multi-volume data sets up to a maximum of 255 volumes.

If the register 15 value is X'FF98', then the code was generated by FDRMOVE, and indicates that the operating system failed to correctly update the catalog Generation Data Group (GDG) extension records. IBM APAR OA24960 corrects this problem.

If the register 15 value is X'FF99', then the code was generated by FDRMOVE, and indicates either that the data set was currently cataloged to an indirect volume or is not cataloged to the input volume. FDRMOVE does not move this data set because it cannot update the catalog.

If the register 15 value is X'5*xxx*', then the code was generated by FDRMOVE, and indicates that "*xxx*" is a VVDS MANAGER return code while trying to update/add an NVR to a VVDS.

If "*status*" is **ALREADY CATALOG VOL=***vvvvvv* then RECAT was not specified, and the output data set was already cataloged to a volume "*vvvvvv*" other than the output volume (or the input volume for a MOVE). For a multi-volume data set, "*vvvvvv*" is the volume in the catalog entry that corresponds to the volume sequence number of the portion of the data set that has just been processed.

If "*status*" is **NAME IS AN ALIAS** then the data set name being cataloged was an alias of another data set in the catalog so FDR did not update it.

Action: FDRMOVE may or may not have moved the data set. Determine the cause of the error and take appropriate action. If desired, contact INNOVATION Technical Support for assistance.

FDR336 DYNAMIC ALLOCATION ERROR COMP=*cc*, CODE=*nnnn*, INFO=*iiii*, DDNAME=*ddname*

Reason: FDRPAS attempted to dynamically allocate (DYNALLOC) a DASD volume but the allocation failed. "*cc*" is the return code in R15, "*nnnn*" is the dynamic allocation reason code, and "*iiii*" is the dynamic allocation information code. For z/OS, these codes are documented in the IBM manual *z/OS V1R12.0 MVS Authorized Assembler Services Guide* (SA22-7608). They can also be found in Appendix A of the ISPF online HELP.

Action: This volume is bypassed.

FDR341 VTOC CONVERT ROUTINE ERROR COMP=*ccc*

Reason: When swapping to a larger or smaller DASD, FDRPAS attempted to allocate a dummy data set with the DOS flag set in the VTOC in order to invoke the DOS VTOC conversion routine to recalculate the free space on a volume. This allocation failed with decimal return code "*ccc*"; these codes are documented under "allocate" in the IBM z/OS *DFSMSdfp Diagnosis GY27-7618-13*. A "*ccc*" of 172 indicates that your security system rejected the dummy data set (that begins with "FDRABR.V*volser*"). A "*ccc*" of 176 or 180 indicates that a user-provided DADSM exit rejected the request.

Action: The volume free space may not be accurate. You must allocate a data set to correct the free space.

If the volume had an active Indexed VTOC (VTOCIX) you must run ICKDSF `BUILDIX` to rebuild it.

FDR343 SMS CONSTRUCT ERROR VOL=*vvvvvv* STORGRP=*storgrp* REASON=*reason*

Reason: FDRMOVE tried to get information about an SMS storage class) but the request failed. "*vvvvvv*" is the volume involved (if known). "*reason*" is the error text:

SMS ERROR COMP=*xxxx* CODE=*ccccc* – The IBM SMS interface failed the query with return code "*xxxx*" (usually 0008) and SMS reason code "*ccccc*". You can interpret the reason code by looking up IBM message **IGD***ccccc* in IBM message manuals.

STORGRP CONSTRUCT NOT FOUND – The specified storage group name was not found by SMS.

STORGRP NOT FOUND OR NO VOLUMES – The storage group is inactive or contains no active volumes. Determine which storage groups and volumes are available using ISMF or the operator command:

```
D SMS,STORGRP(ALL),LISTVOL.
```

STORGRP CONTAINS NO ONLINE VOLUMES – The specified storage group contains no volumes online to the operating system.

Action: FDRMOVE fail with a control statement error.

FDR345 FILTER ERROR REASON=G - ALL ENTRIES EXCLUDED OR DUPS - ENTRY=*dsname*

Reason: No data sets on the searched volumes matched the selection criteria of one or more `SELECT` statements. This can also occur if all the data sets in question are already at the target location; usually from a MOVE operation being run twice with the same selection criteria.

FDR346 FILTER SELECTED *nnnnnnnn* ENTRIES in *sssss.ss* SECONDS

Reason: A filter was used to select entries from the system catalogs. This form of the message documents the number of entries selected by the filter and the elapsed time in seconds required to complete the search.

FDR378 RENAME FAILED COMP=*cccc* CODE=*code* FROMDSN=*tempname* NEWDSN=*permname* CLUSTER=*clusname*

Reason: On an FDRCOPY MOVE or RESTORE of a VSAM cluster to the same name, the output cluster and its components were initially created with temporary names (an index level starting with "**T**" is inserted). After the input data set was deleted, FDRCOPY attempted to ALTER the temporary names back to the original cluster and component names but an error occurred. "*cccc*" is the return code and "*code*" is the reason code from ALTER; look up these codes under IBM message **IDC3009I** in the IBM message manuals.

If "*cccc*" is 04*nn*, then "*nn*" is the return code from a RENAME SVC; if it is 05*nn*, then it is a return code from the SMS RENAME service.

"*tempname*" is the temporary cluster or component name, which FDRCOPY was trying to rename back to "*permname*". "*clusname*" is the original name of the cluster if the rename failed on a component.

Action: If an **FDR378** was not issued for the cluster itself, it was renamed successfully and the cluster can be used under its original name.

If an **FDR378** was issued for the cluster, the cluster was not renamed. It can be used under its temporary name "*tempname*", but you probably want to check the error codes and ALTER the cluster to its original name if possible.

If **FDR378** messages were issued for one or more components, those components were not renamed. The cluster is usable, but you may want to check the error codes and ALTER the component to its original name if possible.

If needed, contact INNOVATION Technical Support for assistance.

FDR990 INSTALLED INNOVATION TRIAL WILL EXPIRE ON *yyyy.ddd*

Reason: The expiration date of a trial version of FDRPAS has been extended successfully. It now expires on date "*yyyy.ddd*".

Required JCL:

```
//EXTEND   EXEC PGM=FDREXTND,PARM=xxxx
//STEPLIB  DD DISP=SHR,DSN=fdrpas.loadlib
//SYSLIB   DD DISP=SHR,DSN=fdrpas.loadlib
//SYSDIAG  DD SYSOUT=*
```

The PARM= is supplied by INNOVATION DATA PROCESSING.

FDR997 *subtask* ABNORMALLY TERMINATED VOL=*vvvvvv*

Reason: This FDR subtask has encountered an error from which it cannot continue. A user ABEND is being issued.

Action: A message detailing the error is printed.

FDR998 FDRPAS COMPLETED WITH ERRORS VOL=*vvvvvv*

Reason: FDRPAS completed but there were diagnostic messages.

Action: Previous message(s) describe the error; see those messages for further details.

FDR999 FDRPAS SUCCESSFULLY COMPLETED

Reason: FDRPAS ran to completion without errors.

390.4 ABEND CODES

Any of the programs and utilities in the FDR family may ABEND (abnormally end) with any of the following user ABEND codes. In many cases, a diagnostic message is printed before the ABEND, so look up any error messages that were printed first. If no message was printed that relates to this ABEND, then read the explanation below. Contact INNOVATION Technical Support if you need assistance understanding or resolving the error.

U0100 OPEN ERROR TRYING TO OPEN A DASD DCB

Reason: Usually proceeded by an **FDR324** message. Check the job log for IBM messages that may indicate the reason for the error.

U0101 MAXIMUM I/O ERRORS EXCEEDED ON A DIRECT-ACCESS DEVICE

Reason: A DASD I/O error has occurred during the FDRPAS copy. The swap is terminated.

U0103 DISK DEVICE TYPE NOT SUPPORTED OR VTOC TOO BIG

Reason: The type of DASD being backed up or restored is not among those supported by FDRPAS or an internal table was exceeded because the VTOC is too large.

U0105 FDRPAS I/O INTERCEPTS FAILED TO INSTALL

Reason: A time-out occurred while trying to install the FDRPAS I/O intercepts on a source volume.

U0107 IEEVARYD DID NOT COMPLETE

Reason: FDRPAS invoked the IBM service IEEVARYD for the target device after a swap, but it did not complete. Contact INNOVATION Technical Support for assistance.

U0108 STORAGE FRAGMENTATION

Reason: When processing a DASD with more than 10017 cylinders, FDRPAS could not find a contiguous storage area for a control record. Contact INNOVATION Technical Support for assistance.

U0300 MAXIMUM FOR COUNT FIELD ERRORS ON DIRECT-ACCESS DEVICE EXCEEDED

Reason: Possibly invalid data was encountered on an FDRPAS source volume. The swap is terminated.

U0301 MAXIMUM FOR INVALID RECORD ZEROS ON A DIRECT-ACCESS DEVICE EXCEEDED

Reason: Possibly invalid data was encountered on an FDRPAS source volume. The swap is terminated.

U0401 SYSIN DD STATEMENT ERROR

Reason: SYSIN DD statement missing or incorrectly specified or I/O error on SYSIN data set.

U0402 SYSPRINT/SYSPRIN*n*/ABRMAP/SYSMAP DD STATEMENT ERROR

Reason: One of the above DD statements is missing or incorrectly coded or I/O error occurred processing the data set. There must always be a SYSPRINT DD statement, and there must be a SYSPRIN*n* DD statement for each backup TAPE*n* DD statement when using ATTACH or ABR.

U0502 ONE OR MORE CONTROL STATEMENTS ARE IN ERROR

Reason: One or more of the control statements specified are not recognized by the program being executed.

U0600 REQUIRED DD STATEMENT IS MISSING OR IN ERROR

Reason: A message is always printed with the DDNAME and reason.

U0609 DYNAMIC ALLOCATION ERROR

Reason: Message FDR336 shows the specific cause.

U0612 EXIT ERROR

Reason: A parameter passed back by a user-written security exit was invalid.

U0650 DISK*xxxx* DD STATEMENT OPEN FAILED

U0801 RESTORE CANCELLED BY OPERATOR OR SECURITY FAILURE

Reason: This ABEND can also be issued for other errors as described by messages in the listing. For an ABR `SNAP` or `SPLIT` with `DSNENQ=`, a data set was found to be in use by another task (message FDR158) so the SNAP or SPLIT is canceled. Specify `ENQERR=NO` on the `SNAP` or `SPLIT` statement if you want to create the point-in-time backup anyway.

U0802 INVALID COMPLETION CODE FROM A USER EXIT

Reason: A parameter passed back by a user security exit was invalid.

U0888 A NON-TERMINATING ERROR OCCURRED

Reason: One or more FDRPAS operations abnormally terminated or ended with diagnostic messages, but the errors were not severe enough to prematurely terminate the entire step. This ABEND is issued to call attention to the errors. Examine the printout for the error messages causing this ABEND.

U0901 ERROR EXECUTING THE FDR TRIAL EXTENSION PROGRAM, FDREXTND

U0902 ERROR EXECUTING THE FDR TRIAL EXTENSION PROGRAM, FDREXTND

U0995 ABEND ESTAE *timeout*

Reason: When the SWAP task of an FDRPAS execution ABENDs (including because of a `CANCEL (C)` command), the ESTAE routine activates cancel protection for two minutes to allow time for proper cleanup. During this time, if a second `CANCEL (C)` command is entered, it is ignored. After two minutes, if termination has not completed, the `CANCEL (C)` command is re-enabled; also, the particular subtask that was in the process of terminating is ended with an immediate **U0995** ABEND.

U0996 *Diagnostic* ABEND

Reason: This is an intentional ABEND activated by a debugging feature (`ABENDM=`).

THIS PAGE INTENTIONALLY LEFT BLANK

Reader's Comment Form

FDRPAS, FDRMOVE, and FDRERASE
User Documentation V5.4 L80
INNOVATION DATA PROCESSING

If you have comments on this manual, including:

- ❒ Errors in the text or typographical errors
- ❒ Clarity
- ❒ Suggestions for improvement in the manual
- ❒ Suggestions for improvement in the product
- ❒ Any other comments

Please complete this form and fax it to INNOVATION DATA PROCESSING at 973-890-7147 (in Europe you may fax it to your local INNOVATION DATA PROCESSING office as shown on the front page of the manual). You may also e-mail your comments to INNOVATION DATA PROCESSING at: support@fdrinnovation.com (be sure to identify the manual name in the message).

Your name:	
Company name:	
Mailing address:	
E-mail address:	

Hardware/Software Profile

Before you begin testing or using FDRPAS, FDRMOVE, or FDRERASE, INNOVATION DATA PROCESSING would appreciate it if you would complete the profile of your hardware and software on this page and fax it to us at 973-890-7147. This saves time for you and us when we provide guidance, assistance, and problem resolution.

Your name: _____

Company name: _____

Mailing address: _____

E-mail address: _____

1	Please list your processors/models, e.g, IBM 2096-K04	_____	_____	_____
2.	How many LPARs of the following do you run?	z/OS: _____	z/VM: _____	z/Linux: _____
3.	Please list the versions of operating systems in use:	z/OS: _____	z/OS: _____	z/OS: _____
4.	Job Entry System:	❒ JES2	❒ JES3	
5.	Security System:	❒ CA ACF2	❒ IBM RACF	❒ CA Top Secret
6.	Cross-system enqueue product:	❒ IBM GRS	❒ CA MIM (CA MII)	❒ _____

7. Total Terabytes (TB) of DASD data: _____ TB
8. Total number of DASD volumes: _____
9. Please indicate the type and manufacturer of the DASD subsystems in use in your installation:

❒ **IBM DS8000 (2107)**	❒ **IBM DS6000 (1750)**	❒ **IBM 2105 ESS (Shark)**		
Emulated Disks:	❒ 3380	❒ 3390-3	❒ 3390-9	❒ 3390-27
	❒ 3390-54	❒ EAV	❒ 1TB	
Features in use:	❒ PAV	❒ PPRC / Metro Mirror	❒ FlashCopy	❒ Metro/Global Mirror
	❒ Concurrent Copy	❒ Consistentcy Group	❒ Global Mirror (XRC)	❒ Preserve Mirror
❒ **IBM RVA (Ramac Virtual Array)**		❒ **Oracle StorageTek SVA or V2Xf**		
Emulated Disks:	❒ 3380	❒ 3390-3	❒ 3390-9	❒ 3390-27
	❒ 3390-54	❒ EAV	❒ 1TB	
Features in use:	❒ SnapShot	❒ PPRC / MetroMirror		
EMC Symmetrix:	❒ **DMX**	❒ **VMAX**		
Emulated Disks:	❒ 3380	❒ 3390-3	❒ 3390-9	❒ 3390-27
	❒ 3390-54	❒ EAV	❒ 1TB	
Features in use:	❒ FlashCopy	❒ PAV / HyperPAV	❒ SRDF	
	❒ Concurrent Copy	❒ Consistency Group	❒ TimeFinder	
Hitachi:	❒ **9900**	❒ **USP / VSP**		
Emulated Disks:	❒ 3380	❒ 3390-3	❒ 3390-9	❒ 3390-27
	❒ 3390-54	❒ EAV	❒ 1TB	
Features in use:	❒ FlashCopy	❒ PAV / HyperPAV	❒ HRC / TrueCopy	❒ PPRC
	❒ NanoCopy	❒ Concurrent Copy	❒ ShadowImage	❒ Consistency Group

Other disk subsystems: _____

10.	Do you use?	❒ IBM's HyperSwap	❒ EMC's AutoSwap	❒ Remote Copy
11.	What (if any) DASD subsystems are you migrating to?	Mfg: _____	Model: _____	

12. Comments / Questions:

_____

_____

_____

If you have any questions or comments, please e-mail support@fdrinnovation.com